ISAAC KERLOW

THE ART OF 3D COMPUTER ANIMATION AND EFFECTS

FOURTH EDITION

The Art of 3D

Computer Animation and Effects

The Art of 3D

Computer Animation and Effects

Fourth Edition

Written and Designed by

Isaac Kerlow

WILEY

John Wiley & Sons, Inc.

This book is printed on acid-free paper. ♾ Typeset in 10/11 Columbus Monotype and Syntax.

Published by John Wiley & Sons, Inc., Hoboken, New Jersey
Published simultaneously in Canada

Library of Congress Cataloging-in-Publication Data
Kerlow, Isaac, 1958-
The art of 3D computer animation and effects / written and designed by Isaac Kerlow. 4th ed.
p. cm.
Includes index.
ISBN 978-0-470-08490-8 (pbk. : alk. paper)
1. Computer graphics. 2. Computer animation. -3. Three-dimensional display systems. I. Title.

T385.K448 2009
006.6'96-dc22 2008047062

Printed in the United States of America

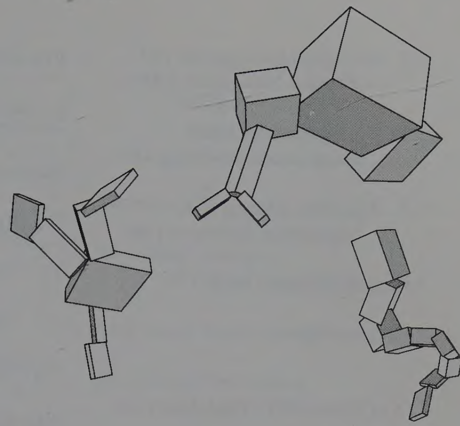
10 9 8 7 6 5 4 3 2 1

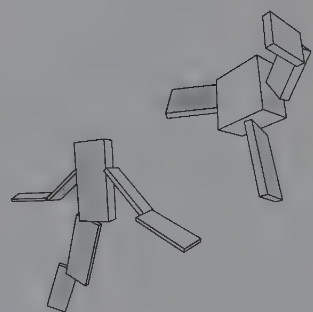
THIS BOOK IS DEDICATED TO ESTHER AND SERGIO KERLOW

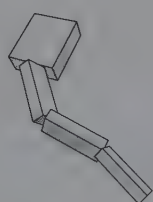
(Pages vii and ix) Black and white line renderings of evolving virtual creatures. (Courtesy of Karl Sims, Thinking Machines Corp.)

Stylized environments frame the comedic action in *Ice Age*. (© 2002 Twentieth Century Fox. All rights reserved.)

Contents







Preface

THE FOURTH EDITION OF THIS BOOK includes many updates and additions that address the new professional realities faced by three-dimensional computer animators, technical directors, and visual effects artists. A lot has changed in this field since this book was first published in 1996. Many of these changes have transformed a field that began as a specialized novelty into a mainstream and mature profession.

Consider the multiple landmark events that have taken place during the first decade of the 21st century. Desktop computers and graphics cards dominate today the production of computer animation, a world previously ruled by high-end (and high-priced) workstations. Computer networks and the Internet, including the development of streaming video on the World Wide Web, have also become ubiquitous during this period. The number of live-action feature films with significant, often stunning, three-dimensional computer animation and visual effects has multiplied. The high-power graphics capabilities of today's game consoles are capable of rendering in real time very sophisticated imagery. High definition digital video is a feasible means of movie production, and all-digital cinema continues to evolve. The quality of work produced by computer animation and film-making students continues to rise worldwide, with student short films winning countless prestigious awards. Computers are used for virtually every single animated movie, game, or TV series produced today, and it seems like the first all-computer-animated feature film was produced a long time ago, before some of this book's readers were born.

Some of the new material that you will find in the book includes:

- Additional new pages focusing on **creative development**, story, visual style, and characters.
- Information on new **production pipelines**.
- An expanded version of the popular **historical timeline** of computer animation, visual effects, and technology, including the recent years.
- Over **300 new color images** with informative captions, including first-time contributions from Australia, Croatia, Czech, Finland, India, Israel, and South Africa and new work from the Hollywood studios, and production houses and independents in Belgium, Canada, China, Denmark, England, France, Germany, Japan, Korea, New Zealand, Norway, Poland, Spain, Switzerland, and the United States.
- New animation **principles of anime**.
- Updated animation chapters with a focus on **blending the traditional animation principles with the digital crafts**.
- **Countless tips** on how to improve the artistic and technical quality of your projects.
- **Updated technical information** on the latest stylistic trends and techniques.
- An **updated website** at ***www.artof3d.com*** containing parts of the book that are available only online, such as the much-read *Ten Tips for Digital Artists*, and other useful resources.

Scope of this Book

IT WOULD REQUIRE AN ENCYCLOPEDIC WORK to fully document and present all of the topics and tech-

niques included in this subject. This book does not pretend to be such a work. A significant effort has been made to create a book that is not bound by the particulars of any specific computer program while, at the same time, offers detailed, **practical information** that goes beyond mere theory. The knowledge contained in this book has been distilled from years of production experience, working and teaching with a variety of software programs, from reading innumerable software manuals, and from practicing and making mistakes. Hopefully, reading this book will help minimize the number of mistakes you make and provide you with the tools to gracefully recover from them. Welcome to artistic production.

How to Read this Book

MASTERING THE ART AND CRAFT of three-dimensional computer animation and effects can be achieved in different ways. Likewise, this book can be read in **multiple ways**.

For those of you who like the **systematic approach** I recommend that you read the book from front to back in sequential order, and refer to the illustrations to complement the knowledge presented in the main body of the text.

For those of you who prefer to **learn by doing** or are too impatient to read the whole book from start to end, it might be best to look at the pictures first. But do not forget to **read the captions** as you look at the images. There is a lot of useful information contained in them. You might want to read some of the sections in the book that complement the step-by-step procedures that you can find in your software manual. This book contains **information rarely found in manuals** or presented by instructors in lab courses. This information is more important than it seems at first because it will help you make the transition from being just a technician (AKA button-pusher) in the software program of your choice to being a creative digital animator or artist.

Using this Book with Software

READ THIS BOOK AS YOU LEARN and experiment with a specific three-dimensional program or by itself without any hands-on work. To those of you who prefer to learn by reading *before* you start the hands-on part of learning, this book offers a **comprehensive introduction** with plenty of theoretical and practical references. Those of you who prefer diving straight into the particulars of a specific computer program—and reading *after* the fact—will find this book complements your experimental approach by offering **clear explanations** in a succinct form. Last but not least, to those of you who learn best by combining the experimental approach with conceptual comprehension, this book offers a **logical progression of topics** that goes beyond the particulars of software manuals.

Book Format

THIS BOOK CONSISTS OF FIVE SECTIONS, each with several chapters. The first section includes a historical overview and general **creative development** and **production issues**. This section deals with a summary of the major creative and technical trends, a short summary of computer animation and visual effects **milestones**, and advice regarding the **production planning** of a project. The introductory section makes you think like a ***historian, film critic***, and ***producer***. Section II goes right into the details of **modeling** three-dimensional objects and environments. The modeling section makes you think like a ***sculptor***. Section III covers many of the most useful **rendering** techniques, both simple and complex. The rendering section makes you think like a ***photographer*** and a ***painter***. The fourth section deals first with many of the issues associated with **telling a story** through **performance,** moving pictures and visual effects: basic screenplay writing, storyboarding, and acting techniques for bringing an animated character to life. The animation section makes you think like an ***actor***, a ***scriptwriter***, and a ***cinematographer***. The last section presents issues involved in **recording** and presenting your work, including sequencing and compositing. This section makes you think like an ***editor*** and a ***magician*** who pulls off the trick right in front of the audience.

What to Expect from this Book

THIS BOOK SEEKS TO INSPIRE AND TO INFORM. It presents the concepts required to understand the **steps and procedures** that lead to the completion of a

fully rendered three-dimensional **computer animation**. Many of the illustrations have been developed to present complex concepts in a way that is clear and **easy to understand**. Considerable effort has also been made so that the book structure and details would be as inclusive as possible: many **software programs and hardware platforms** were tested and reviewed so that readers find a consistent treatment and point of view. The main goal of this book is to provide the reader with a solid foundation by presenting a unique and unusual combination of **technique and creativity**.

What Not to Expect from this Book

This book is not a computer software manual. It is not based on a particular software program. Readers who seek information regarding the detailed operation of specific software or the implementation of specific techniques are advised to consult software manuals. This book is not a general introduction to the use of computers. It is assumed that the reader is already familiar with the basic use of a computer system, or that the reader is in the process of gaining that knowledge elsewhere. Readers who have used computers and who have a basic understanding of the operation of computer systems and are likely to **benefit from this book faster** than readers who have never before used a computer.

About Software Manuals

Software manuals are sometimes tedious to follow and they become obsolete in a flash. The latter is not surprising if we take into account the complexity of software, the constant upgrades of features, and the short production cycles to develop the software and relevant manuals, and turning them into **functional and efficient products** within a competitive market.

Having been both a user and a writer of software user manuals I am familiar with the **frustrations** experienced by both sides. Today's three-dimensional software is so **complex and changing** that we cannot expect a manual to have every single piece of information perfectly digested and updated. My advice is to *take the useful information contained in the software manuals and build on it.*

Some of the nuts and bolts information required to be a proficient digital animator and technician is found in manuals. Some is acquired through **practice**, **trial** and **error**, or from reading books and magazines and consulting with individuals more experienced than oneself. ***Learning requires effort.***

General Principles vs. Specific Techniques

Most of the techniques described throughout this book are available in most of the three-dimensional software programs **available on the market**. But instead of presenting these modeling, rendering, animation, and output techniques in exactly the same ways they are implemented in a specific computer system, we present them by focusing on their **essential features and capabilities**. Specific implementations of techniques—implementations that differ from system to system—are left untouched. Readers who wish to obtain highly specific information regarding a particular computer program mentioned in this book are advised to consult their specific software **reference manuals**.

Acknowledgments

Many of the ideas contained in this book were developed over the years while practicing and teaching the art and craft of three-dimensional computer animation and effects.

I am grateful to my friends and peers who contributed to the process of completing this book, in particular to every single one of the individuals and companies that allowed me to reproduce here images of their amazing work. A special thanks goes to DreamWorks Animation and to Blur Studios for allowing me to have their work on the front cover. Thank you to Singapore's NTU for the sabbatical leave. Thank you to my editor Margaret Cummins, Diana Cisek, and the entire team at John Wiley & Sons for their professionalism.

I hope that students, independent animators, and members of production companies will find this book useful and relevant to achieving their creative goals. Enjoy!

Hollywood, and Orillamar, La Coruña

SECTION 1

Introduction

(Previous page) *Brillia* was created with GROWTH, a recursive algorithm that grows complex life forms. The software repeats simple rules that generate fantastic forms inspired by marine life. (© Yoichiro Kawaguchi.)

1.1.1 *Luxo Jr.* presents a charming view of a small curious lamp who seeks bigger challenges. This short, directed by John Lasseter, was a *tour de force* in three-dimensional computer animation because it was one of the first shorts that tried and succeeded in incorporating many of the traditional principles of animation. (© Pixar Animation Studios.)

CHAPTER 1

Animation and Visual Effects in Context

Summary

THE BRIEF HISTORICAL INFORMATION PRESENTED HERE provides readers with a simple context in which to frame technical and stylistic discussions related to three-dimensional computer animation and visual effects. A summary of events and projects that contributed to the creative and technical development of three-dimensional computer animation and digital visual effects techniques is presented in this chapter.

QUICK INDEX

1.1 A Digital Creative Environment

More than ever before computers have become a part of our life, especially part of our creative, production, and professional lives. They can be found everywhere: they coordinate the flow of information in our banking transactions, they digitize our voices and filter the noise in telephone conversations, they control the fuel injection systems in our cars, and they adjust the settings in photographic and video cameras so that image quality is always optimal. Most of the jobs in the visual professions and trades today require some degree of computer competency. Much of the broadcasting, manufacturing, graphic arts, and entertainment industries have computerized their production. Likewise, many independent artists and design studios develop their work with computers and often deliver it in digital formats.

The transition to increased reliance on computer systems affects many creators and technicians. Large numbers of established visual professionals have retrained to acquire new skills, and young students are eager to learn the secrets and shortcuts. Expectations range from the sensible to the ridiculous. Those who resist the change altogether, for example, are left behind because the world of animation production is changing. Those who are overly enthusiastic often have unrealistic expectations. By finding a balance we can accept the advantages that computer technology has to offer while retaining and developing the knowledge that we inherited from traditional practitioners.

(Top: A *Fearless* blue screen shot. Image courtesy of Menfond Electronic Art & Computer Design Co. Ltd.)

Disney Animated Features without Digital Technology

1937	*Snow White and the Seven Dwarfs*
1940	*Pinocchio*
1940	*Fantasia*
1941	*Dumbo*
1942	*Bambi*
1943	*Saludos Amigos*
1945	*The Three Caballeros*
1946	*Make Mine Music*
1947	*Fun and Fancy Free*
1948	*Melody Time*
1949	*The Adventures of Ichabod and Mr. Toad*
1950	*Cinderella*
1951	*Alice in Wonderland*
1953	*Peter Pan*
1955	*Lady and the Tramp*
1959	*Sleeping Beauty*
1961	*101 Dalmatians*
1963	*The Sword in the Stone*
1967	*The Jungle Book*
1970	*The Aristocats*
1973	*Robin Hood*
1977	*The Many Adventures of Winnie the Pooh*
1977	*The Rescuers*
1981	*The Fox and the Hound*

1.1.2 The Walt Disney Studios created the first animated color feature movie in 1937 and dominated animated feature movies for several decades. These movies were created before the advent of computer animation technology.

Much of today's creation and production of images is indeed performed with the aid of computers. Increasingly, professionals from a wide variety of visual disciplines are working with **digital information**. Some of the traditional visual practices based on drawing, painting, photography, and video techniques are merging with **digital imaging techniques**. A creative environment that used to exist as a collection of totally separate and unrelated disciplines—each with its own tools, techniques, and media—is turning into an environment where visual people use tools and techniques that cross over different media. As a result some of the traditional barriers between visual disciplines no longer have to exist. There is now, for example, great overlap between the fields of animation, graphic arts, broadcasting, and film. The creative **digital environment** has fostered this overlap because computer technology often provides more creative power to visual people. Decades ago, for example, visual professionals used to purchase tools specialized for their professions. These tools were useful for doing the work in their field, but not in others. A photographer, for example, would use a photographic camera to capture reality on film, and a traditional animator would use a pencil and light table to create animated sequences of drawings on paper. Today's photographers and animators, as well as many other creative professionals, share the computer as the tool—loaded with specialized software—to carry on each of their unique tasks.

Animation and Effects in the Predigital Days

For many of us it is difficult to imagine today that only a few decades ago all animation, effects, and entertainment in general was made, distributed, and consumed without any type of computer or digital technology. But this was the case even a few decades ago. One of the first fully three-dimensional computer animated independent shorts, John Lasseter's ***Luxo Jr.***, was released in 1986 (Fig. 1.1.1), and the first fully three-dimensional computer animated feature film, Pixar's ***Toy Story***, was released in 1995 (Fig. 12.1.1).

As we experiment with new modeling, rendering, and animation techniques it is also worthwhile to remember the innovation of many of the animation pioneers, both those who developed character cartoon animation and those who pioneered the use of experimental methods such as collage, cutouts, wax, pinhead shadow, object, and abstract painting animation. Among the former we count the New York and Hollywood animators who delivered gag after gag and amused audiences with the likes of Popeye, Woody Woodpecker, Bugs Bunny, Tom and Jerry, and Mickey Mouse. These animators include Max Fleischer, Walt Disney, Walter Lantz, Tex Avery, Friz Freleng, Chuck Jones, and the many talented animators who worked alongside them in their studios. Among the experimental animators we count the French Léopold Survage and Alexander Alexeieff; the Germans Hans Richter, Oskar Fischinger, and Lotte Reininger; the Canadian Norman McLaren; and the Americans Claire

Parker and John Whitney, Sr. The classic Disney animated movies that popularized the genre were created in the late 1930s and 1940s (Fig. 1.1.2). The well-known twelve principles of animation covered in Chapter 10 were also developed by Disney animators during the same time period, fifty years before computer animation started to be used in their animation studios (Fig. 1.1.3).

As we seek to invent new digital visual effects it is refreshing to remember that in 1939 Hollywood's Academy of Motion Picture Arts and Sciences (AMPAS) created the Special Effects category in their awards competition. Between 1964 and 1971 the category was renamed Special Visual Effects, and between 1972 and 1976 the category of visual effects was renamed as Special Achievement Award, one not necessarily given in a particular year. Since then the category has mostly been called Visual Effects. Winners between 1939 and 2003 are listed in the timelines at the end of this chapter.

If we go back a bit further, just to put three-dimensional computer animation in perspective, we find out that the first kinetoscope parlor opened in New York City in 1894, 101 years before the release of ***Toy Story***. This event was the result of the work of Thomas Alva Edison and his assistant William K. Dickinson to improve the devices for creating images in motion and, above all, the simultaneous recording of sound and motion. Edison and his assistant developed the **kinetoscope**—which means "viewing in motion" in Greek—a closed box in which 50 feet of looped film could be viewed through an opening. The few kinetoscopes equipped with earphones to hear simultaneous music were called **kinetophones**. A few years later on the other side of the Atlantic, during the 1900 Paris Exhibition, a mechanical platform gently rocked by a steam machine presented riders with panoramic views of real and imaginary scenes of the world. These panoramic rides so popular at the turn of the nineteenth century are clearly the ancestors of virtual rides and location-based entertainment. In retrospect it is clear that the kinetoscope spawned many other film viewing and projection systems that fueled the growth of the **seventh art**, cinema, and its animation cousin.

Disney Animated Features with Digital Technology

1985	*The Black Cauldron*
1986	*The Great Mouse Detective*
1988	*Oliver & Company*
1989	*The Little Mermaid*
1990	*The Rescuers Down Under*
1991	*Beauty and the Beast*
1992	*Aladdin*
1994	*The Lion King*, and *The Return of Jafar* *
1995	*Pocahontas*, and *A Goofy Movie* *
1996	*The Hunchback of Notre Dame*, and *Aladdin and the King of Thieves* *
1997	*Hercules*
1998	*Mulan*
1999	*Tarzan*
1999	*Fantasia 2000*
2000	*The Emperor's New Groove*, *Dinosaur*, and *The Tigger Movie* *
2001	*Atlantis: The Lost Empire*, and *Recess: School's Out**
2002	*Lilo & Stitch*, *Treasure Planet*, and *Return To Never Land* *
2003	*Brother Bear*, *Jungle Book 2*,* and *Piglet's Big Movie* *
2004	*Home on the Range*, and *Mulan 2* *
2005	*Chicken Little* (all-3D CG)
2007	*Meet the Robinsons* (all-3D CG)
2008	*Bolt* (all-3D CG)
2009	*The Princess and the Frog*
2010	*Rapunzel* (all-3D CG)

1.1.3 The Walt Disney Studios started using digital technology in the production of its animated features during the mid-1980s. This listing also includes (marked with an asterisk *) a few of the movies produced as "straight-to-video," mostly by the Television Animation division. Some of the straight-to-video titles also had a limited theatrical release.

1.2 The Development of the Technology

Computers, particularly their visual capabilities, are profoundly altering the way in which we create and distribute images. But the powerful computer systems that are so common today—and that everybody takes for granted—have existed for a relatively short period of time.

The ancestors of today's electronic digital computers were mechanical adding machines used to perform repetitive arithmetic calculations. Those early mechanical devices eventually evolved into machines that could be programmed each time they were used to perform different sets of instructions. In the 1940s, electric versions of these computing machines were in operation.

The early computer models were called **mainframes** because all

SIGGRAPH Computer Animation Festival Awards

Best of Show Award

1999	*Bunny*, Chris Wedge
2000	*Onimusha*, Takeshi Kaneshiro
2001	*Values*, Van Phan
2002	*The Cathedral*, Tomek Baginski
2003	*Eternal Gaze*, Sam Chen
2004	*Birthday Boy*, Sejong Park
2005	*9*, Shane Acker
2006	*One Rat Short*, Alex Weil
2007	*Ark*, Grzegorz Jonkajtys and Marcin Kobylecki
2008	*Oktapodi*, Emud Mokhberi, et al.

Jury Honors Award

1999	*Masks*, Piotr Karwas
2000	*Stationen*, Christian Swade-Meyer
2001	*F8*, Jason Wen
2002	*Le Deserteur*, Olivier Coulon, Aude Danset et al.
2003	*Tim Tom*, Romain Segaud and Christel Pougeoise
2004	*Ryan*, Chris Landreth
2005	*Fallen Art*, Tomek Baginski; and *La Migration Bigoudenn*, Eric Castaing et al.
2006	*458nm*, Jan Bitzer et al.
2007	*Dreammaker*, Leszek Plichta
2008	*Mauvais Rôle*, Alan Barbier, Camille Campion et al.

1.2.1 (Above) Recipients of the SIGGRAPH Computer Animation Best of Show Award and the Jury Honors Award.
(Opposite page) Recipients of the SIGGRAPH awards for technical research in computer graphics, including the Computer Graphics Achievement Award, and the Steven A. Coons Outstanding Creative Contributions Award.

their bulky components were housed in large steel frames. During the 1960s two types of computers were developed. **Minicomputers**, smaller and less expensive than mainframes but almost as powerful, were developed in an attempt to bring computers to a wider audience and range of applications. **Supercomputers**, usually bigger and more expensive than mainframe computers, were developed to tackle the most taxing computing projects regardless of the cost and with an emphasis on speed and performance.

Before the mid-1970s the large majority of artists found computers very uninteresting. They were too expensive and cumbersome to operate, and even the simplest tasks required extensive programming. Most models lacked monitors, printers, mice, or graphics tablets. **Microcomputers** with millions of microscopic electronic switches on a single **silicon chip** were developed in the mid-1970s. Some early models of microcomputers, such as the Apple Macintosh, the Amiga, and a variety of Intel-based PC computers, were widely embraced by visual professionals during the 1980s (Fig. 1.2.2). Many of today's powerful microcomputers are small enough to be carried in a bag or briefcase. Those that can fit in a pocket still have limited capabilities for professional image creation, but many are quite good at displaying moving images of different degrees of quality. The supermicrocomputer and the parallel computer were developed during the 1980s and had a great effect on the way visual people use computers. **Supermicrocomputers**—also called **workstations**—are microcomputers built around a powerful CPU that is customized to excel in the performance of a specific task, for example, three-dimensional computer animation. **Massively parallel computers** deal with very complex processing challenges by dividing up the tasks among a large number of smaller microprocessors. Some of these computers may have between a dozen and thousands of processors.

Computer graphics technology was developed in the early 1950s to make visible what was invisible to the human eye. Most of these early applications were related to the military, manufacturing, or the applied sciences and included, for example, flight simulators to train fighter pilots without having to fly a real plane; computer-aided design and manufacturing (CADAM) systems to allow electrical engineers to design and test electronic circuits with millions of components; and computer-aided tomography (CAT) scanners to allow physicians to peek into the human body without having to physically open it. None of the early computer graphics systems was developed for artistic work.

During the 1950s and 1960s, the early years of computer graphics technology, the computer systems and techniques for creating images were rudimentary and very limited—especially by today's standards. During that period very few artists and designers even knew that computers could be used to create images.

During the 1970s and 1980s computer technology became more practical and useful, and a significant number of visual creators started to get interested in using computers. During the 1990s a sig-

nificant drop in the prices of computer systems and an increase in their computing power occurred. This situation encouraged many visual professionals to purchase the technology and to integrate it into their daily professional practices. Professionals from all visual disciplines accepted computer technology as it became more powerful, more practical, and less expensive. Many of the major technical innovations in the area of computer animation and related applications have traditionally been presented at the **SIGGRAPH** annual conference. Sponsored by the Association for Computing Machinery's Special Interest Group in Graphics, SIGGRAPH has been the most influential professional association in the field of computer animation since the 1960s. Figure 1.2.1 lists the international computer animation projects awarded at SIGGRAPH's screenings since the inception of the awards. It also lists some of the computer science and engineering pioneers in the field of computer graphics as recognized by their peers through the SIGGRAPH awards. Some of the research papers and innovations of these technical pioneers can be found in the conference proceedings.

The computer technology necessary for creating three-dimensional imagery and animation has evolved tremendously since the first systems were developed in the 1950s. Within just a few decades the capabilities of hardware and software for creating three-dimensional environments went from simple to highly complex representations that often fool our visual perception.

A complete history of three-dimensional computer graphics technology and creative works remains to be written. However, the information presented in the rest of this chapter summarizes some of the highlights and landmarks. This summary is certainly not exhaustive, and it does not attempt to present a complete and detailed portrait of all the significant events. Instead, it provides a personal account of individual examples and a summary of the major trends.

Technical Developments: 1950s and 1960s

The decades of the 1950s and 1960s saw the development of the first interactive computer systems, which were further improved during the following decade. The field of computer graphics was so new then, and most of the technological innovations from this period did not yield spectacular visual results. These innovations were, however, fundamental in facilitating the impressive developments that would flourish 20 years later.

The first computer to use CRT displays as output peripherals was the Whirlwind computer at the Massachusetts Institute of Technology (MIT) in the early 1950s. This system was used to display the solutions to differential equations on oscilloscope monitors. During the mid- to late-1950s the SAGE Air Defense System of the U. S. Air Force used command-and-control CRT displays on which operators could detect aircraft flying over the continental United States. The SAGE operators were also able to obtain information about the air-

ACM SIGGRAPH Awards

Computer Graphics Achievement

1983	James F. Blinn
1984	James H. Clark
1985	Loren Carpenter
1986	Turner Whitted
1987	Robert Cook
1988	Alan H. Barr
1989	John Warnock
1990	Richard Shoup and Alvy Ray Smith
1991	James T. Kajiya
1992	Henry Fuchs
1993	Pat Hanrahan
1994	Kenneth E. Torrance
1995	Kurt Akeley
1996	Marc Levoy
1997	Przemyslaw Prusinkiewicz
1998	Michael F. Cohen
1999	Tony DeRose
2000	David H. Salesin
2001	Andrew Witkin
2002	David Kirk
2003	Peter Schroder
2004	Hugues Hoppe
2005	Jos Stam
2006	Thomas W. Sederberg
2007	Greg Ward
2008	Ken Perlin

Outstanding Creative Contributions

1983	Ivan E. Sutherland
1985	Pierre Bézier
1987	Donald P. Greenberg
1989	David C. Evans
1991	Andries van Dam
1993	Ed Catmull
1995	Jose Luis Encarnação
1997	James Foley
1999	James F. Blinn
2001	Lance J. Williams
2003	Pat Hanrahan
2005	Tomoyuki Nishita
2007	Nelson Max

Timeline of Intel Processors

1971	The 4-bit 4004 with 2,300 transistors, and 108 KHz clock.
1972	The 8-bit 8008, twice as powerful as the 4004.
1974	The 8080, CPU of the Altair, first personal computer.
1978	The 8086-8088, CPU of the IBM PC.
1982	The 286, first Intel processor that runs software written for its predecessor.
1985	Intel 386, 32-bit chip with 275,000 transistors, 100 times more than the 4004.
1989	The 486, with built-in math co-processor, for graphical interfaces, 25-50 MHz.
1993	First Pentium processor.
1995	Pentium Pro, 5.5 million transistors.
1997	Pentium II, 7.5 mill. transistors and video processing MMX technology, 200-300 MHz.
1999	Celeron, value-oriented.
1999	Pentium III, 9.5 million transistors, 650 MHz to 1.2 GHz.
2000	Pentium 4 debuted with 42 million transistors, real-time 3D rendering, 1.3-1.8 GHz.
2001	Xeon, for high-performance dual-processor workstations.
2001	Itanium, first in a family of 64-bit processors.
2004	Pentium 4, Hyper-Threading technology, at 3.4 GHz.
2005	Pentium extreme Edition 840 dual-processor workstations.
2006	Energy-efficient Core 2 Duo, 291 mill. transistors and Core 2 Extreme.

1.2.2 Some of the most popular microprocessors manufactured by Intel since the early days of microcomputers.

craft by pointing at their icons on the screen with light pens.

During the 1960s various technology-intensive organizations developed the first **computer-aided design and manufacturing (CADAM)** systems. The goal of these early CADAM systems was to make the design process more effective by offering users sophisticated design functions and to improve the organization of the manufacturing process by linking the numerical data that represents an image with other types of information, such as inventory and engineering analysis. One of the first CADAM systems was developed by General Motors, and it included various time-sharing graphic stations for designing cars. Other companies, including Boeing Aerospace, IBM, McDonnell Douglas, General Electric, and Lockheed, developed similar systems.

Early attempts to create computer-generated movies took place in several research institutions. Short pieces of animation were produced at Boeing by William Fetter and Walter Bernhart in the early 1960s. Three-dimensional drawings were plotted on paper and filmed one at a time to produce animations of an aircraft carrier landing. Fetter also modeled the human figure for ergonomic studies related to the design of cockpits. At Bell Laboratories, researchers Michael Noll and Bela Julesz produced various stereo computer animations on film to aid in the study of stereo perception. During this period some of the first animation programming languages were developed, but most of them resulted in programs that ran only in a noninteractive mode.

Only a few commercial companies were involved in computer graphics research during these two decades. Most of the technological developments during this period came out of government-funded academic research laboratories such as MIT's Lincoln Labs.

In the early 1960s, computer graphics were developed to visualize objects and situations that were too costly or just impossible to represent otherwise. Flight simulators, CADAM systems, and CAT scanners were among the pioneering computer graphics systems.

The first interactive system, called **Sketchpad**, was developed in the early 1960s by Ivan Sutherland at MIT. This system allowed users to interact with simple wireframe objects via a light pen. This system made use of several new interaction techniques and new data structures for dealing with visual information. It was an interactive design system with capabilities for the manipulation and display of two- and three-dimensional wireframe objects.

By the mid-1960s the first algorithms for removal of hidden surfaces were developed, and the systems for producing full-color surface-shaded animation in real time were improved. General Electric developed a flight simulator that animated and displayed simultaneously as many as 40 solid objects with hidden surfaces removed and in full color. The Mathematical Applications Group, Inc. (MAGI) in Elmsford, New York, was one of the first companies that offered computer-generated animation of fully rendered polygonal objects in the commercial environment. Their process was named

Synthavision, and its first contracts included simulations for the military and advertising-related projects.

The early three-dimensional computer animation and imaging systems depended on costly mainframe computers that were slow by today's standards. Most of the programs would run only on a specific type of computer and display device, and were not portable to other systems. The use of computer graphic systems during the 1960s was restricted by the high cost and limitations of the hardware involved.

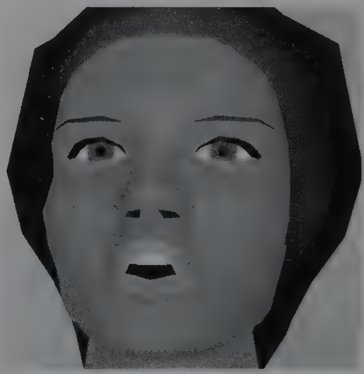
Virtually all of the graphics software from this period was developed in-house. It was not marketed, and it was minimally documented. Most programs were executed in the batch mode, and very few had any interactive features at all. Users had to input their data almost exclusively through the keyboard; other types of input peripherals that encouraged more artistic freedom were just not available. A few computer systems had graphics screens, but most had monochrome alphanumeric CRT screens or just teletype or dot matrix printers.

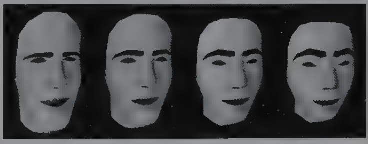
1.2.3a Early example of hand animation. (Courtesy of Ed Catmull.)

Technical Developments: 1970s

The 1970s was a significant decade for the development of computer animation and imaging technology. Many of the basic rendering techniques still in use today were formulated during the 1970s. The minicomputers that became popular during the 1970s were easier to maintain than mainframe computers, and provided significantly more power at a reduced cost. Microcomputer technology was also introduced to the consumer markets in the late part of the decade.

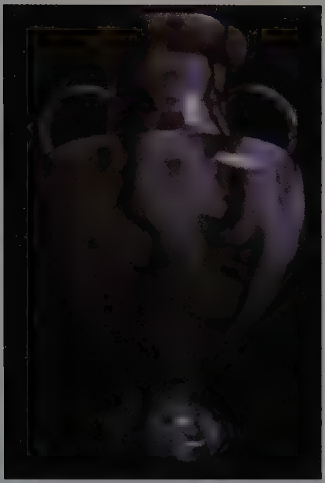
From the point of view of computer hardware, most of the research and production work done during this decade was based on minicomputers. The microcomputer's 8-bit computing power, memory capabilities, and output solution was insignificant when compared to their high-end counterparts (Fig. 1.2.2). But in the videogames arena the new microcomputers greatly contributed to the popularization of computer-generated graphics. A standard configuration of an early 1970s microcomputer included an 8-bit CPU without any graphics co-processors, less than 100 KB of RAM memory, a clock speed of 10 MHz, a low resolution screen with a maximum palette of 8 colors (or slightly higher if dithering was used), and a limited amount of peripheral storage.





1.2.3b Early examples of face animation. A 1972 model (top) used simple expression interpolation. A later sequence used shape interpolation. (Images courtesy of Frederic Parke.)

During this decade the University of Utah became a primordial force and a center of innovation in three-dimensional computer graphics research. Under the guidance of David Evans, co-founder of Evans & Sutherland, the Department of Computer Science at the University of Utah produced a distinguished roster of Ph.D. students. Many of them developed a large number of the major technical contributions of the decade, such as the original versions of polygonal, Gouraud, and Phong shading; image and bump texture mapping; z-buffering; the subdivision and the painter's algorithms for hidden line removal; antialiasing methods; and hand and facial computer animation (Fig. 1.2.3).



1.2.4 Vase rendered with stochastic textures. (Image courtesy of Ken Perlin.)

1.2.5 Developed jointly by Toyo Links and Osaka University, the muscle movement in *Bio-Sensor* was modeled with metaballs and rendered with ray tracing techniques. (Director: Takashi Fukumoto. Technical Director: Hitoshi Nishimura. © Toyo Links, 1984.)

Technical Developments: 1980s

It was during the 1980s when computer graphics technology leaped from being a curiosity into becoming an area of proven artistic and commercial potential. Technologically speaking, this decade started with the uneven coexistence between powerful minicomputer systems and 8-bit microcomputers. But the decade ended with a new breed of powerful 32-bit microcomputers and 64-bit RISC (reduced instruction set computer) graphics workstations at the forefront, and with minicomputers in the back seat. Silicon Graphics Inc., the company that pioneered visual workstations with its **Geometry Engine**, was started by James Clark in 1982. Commercially speaking, this decade started with a handful of companies that pioneered the production of three-dimensional computer animation and imaging. These companies included Digital Effects and MAGI on the East Coast and Robert Abel Associates and Information International, Inc. (III) on the West Coast. These companies operated exclusively with software developed in-house and with much custom-built graphics hardware. The 1980s concluded with the closing of all of the pioneer production houses—or at least of their production divisions—and with the creation of a new group of smaller, leaner, and more market-oriented firms that operated mostly with off-the-shelf hardware and with a mixture of custom and off-the-shelf software.

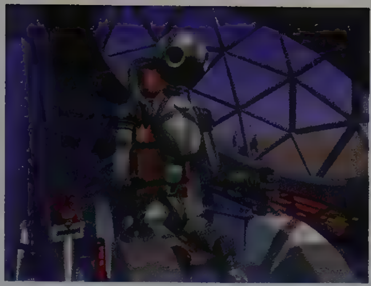
The bulk of the software research and development during this decade was spent in refining the modeling and shading techniques inherited from the 1970s. Ground was broken with new rendering approaches such as radiosity and procedural textures, and with the development of the first generation of solid user-friendly computer-human interfaces for three-dimensional computer animation and imaging software. The **RenderMan** shading language was released by Pixar in 1988. Some of the software companies that pioneered the high-end tools for three-dimensional computer animation and visual effects production were founded during this period. In 1981 Wavefront opened in Santa Barbara, California. In 1982 Alias opened in Toronto, Softimage in Montreal, and Mental Images in Berlin in 1986. Side Effects Software opened in Toronto in 1987.

Some of the leading academic centers in North America involved in three-dimensional graphics research during this period included Cornell University (radiosity rendering techniques), the Jet Propulsion Laboratory at the California Institute of Technology (motion dynamics), the University of California at Berkeley (spline modeling), Ohio State University (hierarchical character animation and inverse kinematics), the University of Toronto (procedural techniques), the University of Montreal (character animation and lip syncing), and New York University (procedural textures, Fig. 1.2.4). Significant original research efforts also took place at the University of Tokyo and Osaka University (modeling with blobby surfaces, Fig. 1.2.5), and the University of Hiroshima (radiosity and lighting). A few research centers and private companies invested significant resources

in the development and production of shorts that pushed computer graphics technology to its limits (Figs. 1.2.6 and 1.2.7). Government-sponsored research centers also developed pioneering simulation techniques. Figure 1.2.8 shows a landmark simulation of natural forces. Most of the three-dimensional software products that were commercially available during the first half of this decade lagged behind the exciting work done in research institutions. This was partly due to the lack of capital investors who believed in the commercial potential of the technology. It was also due to the difficulty of implementing computing-intensive techniques with off-the-shelf hardware systems that were not quite as fast as needed and perhaps a bit overpriced.

Some of the hardware research during the 1980s focused on the development of more powerful general-purpose microprocessors and special-purpose graphics processors, and techniques for the high-speed transfer of visual data. A standard midrange computer system for three-dimensional computer animation production during the 1980s, for example, consisted of a 32- or 64-bit microcomputer or supermicrocomputer with one or several graphics processors, clock speeds higher than 50 Mhz, several dozens of megabytes for RAM memory, and extensive peripheral storage.

In terms of output standards, few of the production companies at the beginning of the decade were capable of first-generation output to videotape. Most of the high-end work was output to film first and then transferred to videotape. By the end of the decade, however, video output established itself as a common output method for computer-generated animation.

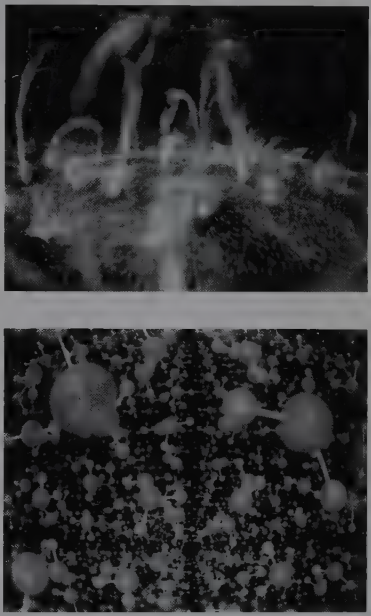


1.2.6 Mechanical ant from the 1984 trailer for *The Works*. (Design, animation, and modeling by Dick Lundin. Robot model by Ned Greene. Images courtesy of Ned Greene and NYIT Computer Graphics Lab.)

Technical Developments: Early 1990s

The first half of the 1990s witnessed a major move toward smaller and/or considerably more powerful computer systems. Virtually all of the low-end microcomputers currently in production are based on 32-bit microprocessors, whereas the powerful microcomputer models are centered around 64-bit CISC and RISC processors. A considerable number of models with different features were targeted at different segments of the market, especially systems with multiple processors, and computer systems were sold in a modular form. Supermicrocomputers, or workstations, kept increasing in power while decreasing in price, or remained at the same price level but with additional features.

Research and development was mostly centered around issues of efficiency, cost, and ease of use. With the midrange hardware systems being powerful enough for most creative needs, a lot of energy and time was spent in optimizing the software techniques. Taking small solid steps took precedence over making large groundbreaking advances. Computer-human interface issues also took precedence in this buyer's market. Users of three-dimensional software became increasingly sophisticated and demanding, and enjoyed new levels of creativity and computing power. Two additional trends of this period



1.2.7 *The Universe* was the world's first stereoscopic (stereo 3D) computer animation rendered and output on 70 mm 15-perf film for projection on an IMAX dome screen. It was created for the Fujitsu Pavillion at Tsukuba Science Expo '85. (© 1985 Fujitsu/Dentsu/Toyo Links/IMAX.)

1.2.8 In this simulated snapshot of a severe storm (top) the yellow-gold regions represent small cloud drops and ice particles, and the blue region represents large water drops. The surface grid lines are 10 km apart and the darkened area indicates the horizontal integration domain surface of the storm. The orange-red ribbons (bottom) represent the tracer particles in that rise through the depth of the storm in the updraft, and the blue ribbons represent tracers that eventually fall to the ground in the downdraft. The spatial resolution of the data used in the simulation was 2 km horizontally and .75 km vertically. (Image from "Study of a Numerically Modeled Severe Storm." Courtesy of the National Center for Supercomputing Applications.)

include the rebirth of the electronic game industry (and the growth in jobs, volume, and quality associated with it), as well as the fact that, overall, the computer industry became friendlier and less technical as it tried to mass-market its products to the consumer market.

Technical Developments: Late 1990s

During the second half of the 1990s the worlds of computer animation and visual effects production were impacted by the many changes and technological advances that took place in the computer hardware and software industries. Some of the more influential events include the popularization of the **Windows NT** and **Linux** computer operating systems. This trend evolved to a point where even SGI, the computer company formerly known as Silicon Graphics, Inc. and a traditional stalwart of the **UNIX** operating system, began offering NT-based computers in 1999. CPU clock speeds used for production continued to move up, with speeds of 400 and 500 MHz becoming common. Intel and its Pentium line of processors became a significant player in production environments where years earlier they had been dismissed as lightweights. Powerful graphics co-processors designed to accelerate the speed of three-dimensional computations also continued to evolve. Some ended up being bundled on PC motherboards, and others continued to be sold as add-on graphics cards. Computer networks became vital to digital production due to the popularization of rendering farms and production in multiple locations. By the end of the decade the use of company **computer intranets** for communications and file transfer and management became the standard practice at most leading centers of digital production.

Major advances in the video industry also had an impact on computer animation and visual effects production. **Digital video** became a reality in the mid-1990s, and by the end of the decade different types of productions had been realized in the new medium. Late in the decade Sony introduced a **high definition digital video** camera, and in 1999 the company announced that a 24-frame-per-second version (24P HD) was in development. In the same year the filmmaker George Lucas made statements about his plans to shoot live action with such a camera in the second *Star Wars* prequel. The success of 1999 independent films like *The Blair Witch Project* and the Danish *The Celebration* popularized the acceptance of digital video for mainstream production. These two live action films were shot on digital video (the former used only available light) and transferred to 35 mm film for theatrical release. The increasingly widespread availably of 24P HD digital video promises to simplify some production issues in the area of visual effects.

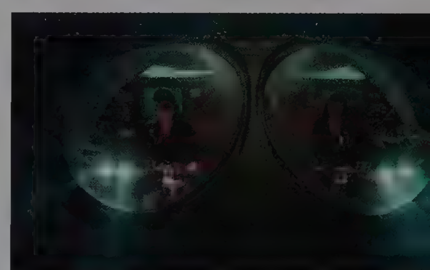
The advent of **digital movie projectors** was another significant development of the late 1990s because it pointed to the fact that one day photographic film might no longer be the dominant medium for motion pictures and animation, not only for home use but also for theatrical releases. Texas Instruments became an early

leader in the field of digital projectors when it introduced its first DLP model in 1998.

The great popularity and growth experienced by **computer games** and **platform games** translated into many jobs for three-dimensional computer animators, who created hundreds of real-time and prerendered games for personal computers and new game platforms such as Sony's **PlayStation**, the **Nintendo 64**, and Sega's **Dreamcast**. The arcade version of *Virtua Fighter 3* released by Sega in 1996 brought arcade games to a new level of polygon real-time rendering performance. But the new home console game systems continued to become more sophisticated. The game industry during the 1990s wrapped up with a great demand for computer animators and several exciting games.

In the area of software tools, there were several significant developments during the late 1990s. The modeling technique of subdivision surfaces allows users to build three-dimensional models with variable geometry resolutions throughout the model. This technique was developed at the University of Washington and then perfected at Pixar, where the 1997 award-winner *Geri's Game* became the test-bed for this modeling technique (Fig. 1.2.9). The technique of image-based rendering, refined at The University of California at Berkeley, facilitates the reconstruction of three-dimensional environments based on photographic references taken on location (Figs. 6.8.1–6.8.3). Nonphotorealistic rendering is used to represent three-dimensional geometry with a two-dimensional look. Applications of this approach are of great interest to technical illustrators and to

1.2.9 The 1997 computer-animated short *Geri's Game* was a technology test-bed for subdivision surfaces modeling techniques and for clothing dynamics. Rendering was done with the RenderMan shading language. (© Pixar Animation Studios.)



1.2.10 *Spy Kids 2* was one of the first movies to be recorded on high-definition (HD) video. (© 2002 Hybride. Images courtesy of Dimension Films.)

Game Platforms General Specifications

Sixth Generation

Microsoft Xbox (2000)
Rendering: 125 million triangles per second (TPS)
CPU: 32-bit Pentium III, 733 MHz
GPU: Nvidia, 250 MHz
RAM: 64 MB
Storage: 6.4 GB

Nintendo GameCube (2000)
Rendering: 12 TPS
CPU: 32-bit IBM PowerPC Gekko, 485 MHz
GPU: ATI/Nintendo, 162 MHz
RAM: 43 MB
Storage: 1.5 GB

Sony PlayStation 2 (2000)
Rendering: 66 TPS
CPU: 128-bit RISC Emotion Engine, 300 MHz
GPU: 147.5 MHz
RAM: 40 MB
Storage: 4.7 GB

Seventh Generation

Microsoft Xbox 360 (2005)
Rendering: 500 million TPS
System Performance: 1 TFLOPS
CPU: 3 PowerPC-based cores 3.2 GHz
GPU: Custom ATI, 500 MHz
RAM: 512 MB of GDDR3
Storage: 20–120 GB HD

Sony PlayStation 3 (2006)
Rendering: 275 million TPS
System Performance: 2 TFLOPS
CPU: PowerPC Cell, 3.2 GHz
GPU: Custom Nvidia, 550 MHz
RAM: 256 MB XDR Main, and 256 MB GDDR3 VRAM
Storage: 20–120 GB HD

1.2.11 Specs of three popular sixth generation game platforms at time of release, and the Xbox 360 and the PlayStation 3.

those looking for ways to visually integrate traditional animation and three-dimensional computer animation (Figs. 6.9.1–6.9.4). The simulation of water and gas dynamics and brittle matter also gained ground during this period (Figs. 5.5.10, 12.3.3, and 12.3.4).

Technical Developments: Early 2000s

Hardware continued its march toward smaller, faster, and cheaper. Graphics boards with powerful GPUs made hardware rendering a reality, both for professional applications and home entertainment. The game industry benefited from this increased computing power for the playback of real-time three-dimensional computer animation. Powerful home game consoles proliferated (Fig. 1.2.11). In 2000, for example, Sony introduced the **PlayStation 2**, a system built around a 128-bit processor that draws 2 million polygons per frame, which is about the geometry resolution of an average scene in the 1995 *Toy Story* animated feature film. Microsoft introduced the **Xbox** powered by an Nvidia graphics card capable of drawing up to 125 million polygons per second, and Nintendo introduced the **GameCube**. The competition for revenues between the computer and platform games and movie box office continued.

Processors for PCs moved to speeds closer to and beyond 2 gigahertz (GHz). Intel's Pentium 4 Processor, for example, debuted in 2000 with an initial speed of 1.5 GHz—compare that to Intel's first microprocessor, the 4004, which ran at 108 KHz! If automobile speed had increased similarly over the same period, you could now drive from San Francisco to New York in about 13 seconds. AMD introduced their 64-bit processor in 2003. 64-bit processors have attracted interest from software developers and Mental Ray is an early 64-bit renderer. In the area of operating systems, relative newcomer Linux became the dominant standard for high-end production, while **Mac OSX** gained a fair amount of attention and use because of its UNIX-like features.

High-definition video (HD) continued to develop with the successful completion of all-digital productions such as *Star Wars: Episode II* and *Spy Kids 2* (Fig. 1.2.10). Different flavors of HD cameras continued to be introduced, most notably Thompson's **Viper FilmStream** in 2002, prototypes by Olympus and JVC, as well as Sony's 1080p HD camera system, the **F950 CineAlta** with a 2/3 in. sensor and 4:4:4 full-RGB bandwidth, and the improved HDCAM-SR tape format with 4:2:2 10-bit color bandwidth. At the same time the use of digital dailies and the digital intermediate process gained converts in the worlds of movie and TV post-production. File formats for compressing and decompressing files sent via computer networks developed significantly. These formats, known as **codecs**, are especially useful for video streaming and resulted in new versions of Apple's Quicktime, Microsoft's Media Player 9, and a few others based on the MPEG-4 video compression standard (read Chapter 15 for more information on compression).

The proliferation of DVD players in the home video consumer

market during the early 2000s helped fuel a massive distribution of the computer-animated and visual effects "hits" of the period. A *Hollywood Reporter* survey in 2003, for example, indicated a 65 percent increase in DVD purchases between 2001 and 2002. VHS sales declined an estimated 29 percent during the same period.

Technical Developments: Late 2000s

The seventh generation of game platforms was released in the middle of the decade with impressive specs. In 2005, for example, Microsoft introduced the **Xbox 360** capable of rendering 500 million triangles per second (Fig. 1.2.11). Nintendo released the **Wii** also in 2005, based on a RISC IBM PowerPC processor and an ATI Hollywood graphics card, and Sony followed up with the **PlayStation 3** in 2006. Computer and graphics cards manufacturers continued to offer faster and more powerful systems. In 2006 Nvidia graphics card manufacturer started to distribute a free version of their hardware renderer **Gelato** with native 64-bit support. Also in 2006 CPU manufacturer AMD merged with graphics card manufacturer ATI Technologies, and scheduled a new product that merges a CPU and a GPU for 2009 release. Nvidia has also announced a planned integration of a CPU and a GPU on a single chip. In 2006 Apple Computer switched to Intel CPUs for its line of Macintosh computers and servers.

On the digital cinematography front a variety of new high-definition cameras were introduced or improved. Many of these non-film-based cameras are increasingly used for recording commercials or live action features, particularly those with a large number of visual effects. The **Red One** camera is notable for its high resolution and its relative low price. The Red One was released in 2007 with a 24.4 x 13.7 mm 12-megapixel sensor delivering a 4096 x 2304 pixel image. Effects movie *Jumper* (2008), and *Guerrilla* and *The Argentine* (2008), both directed by Steven Soderbergh, were shot with the Red One (Fig. 13.12.5). In 2004 Panavision introduced the **Genesis** camera with a single 12.4-megapixel sensor and 4:4:4 color bandwidth. It was first used in 2006 to shoot effects feature *Superman Returns*, *Scary Movie 4*, and *Apocalypto*, and in 2008 to record *Asterix in the Olympic Games*. In 2007 Dalsa introduced the **Origin II** camera with a resolution of 4046 x 2048 pixels and the ability to output uncompressed 16-bit image files, yielding very wide dynamic range. The James Bond movie *Quantum of Solace* (2008) includes a visual effects shot captured with eight Dalsa cameras shooting simultaneously and with synchronized shutters. Arriflex introduced in 2005 its **D-20** camera featuring a 1920 x 1080 pixel resolution in 4:4:4 10-bit format. Vision Research offers the **Phantom HD** high-speed camera also capable of 2K resolution and frame rates ranging from 1 to 1000 fps at 14-bit depth per channel.

On the consumer front YouTube launched in 2005 as a video sharing website that has allowed animation and effects professionals

1.3.1 Still frames from the Voyager space mission simulations. Closest approach of Voyager 1 to Jupiter's moon Callisto (top). The field of view of the onboard narrow-angle camera is projected as a series of squares on the moon showing the planned images that were sent back to Earth. (Second-Fourth) Simulations of Voyager 2: Closest approach to Jupiter's moon Europa with texture from photographs sent back by the earlier mission; crossing of the equatorial ring plane of Jupiter; flight over the rings of Saturn, made artificially brighter for visualization purposes. (Images are courtesy of Jim Blinn, JPL/Caltech.)

to easily showcase their work, and has also brought a lot of animation to mainstream viewers. Google's Image Search became a useful visual research tool and in 2005 it reached 1.1 billion images indexed. Google acquired YouTube in 2006. The Blu-ray Disc became in 2008 the high-capacity de facto standard for DVD format: it can store up to 50 GB in dual layer mode.

1.3.2 Landscapes from *Vol Libre* created with fractal techniques. (© 1980 Loren Carpenter.)

1.3 Visual Milestones: 1960–1989

It is both refreshing and illuminating to view, enjoy, and analyze the visual works that became creative milestones in the development of three-dimensional computer animation and imaging. By analyzing these works we can learn about all the computer animation techniques and styles that have evolved into what this field is today.

Useful sources for learning more about computer animation and visual effects include the **SIGGRAPH Video Review** DVDs, issues of the *Cinefex* journal of visual effects, and the winners and runner-ups of the American **Academy of Motion Picture Arts and Sciences (AMPAS)** awards in the Best Visual Effects and Best Animated Feature, and Best Animated Shot categories (see listings in the timelines at the end of this chapter). Visit the **www.artof3d.com** website for links to these sources, as well as other useful computer animation resources and links.

Visual Milestones: 1960s

During most of the 1960s the computer was—in the opinion of most visual artists, critics, and spectators—too cold and technical to be involved in the creation of artistic projects. Similar prejudices about technology arose in the nineteenth century when machines were introduced on a massive scale to the industrial world. Many turn-of-the-century painters feared the new technology until they learned how to use it and became creative with it. During the 1960s the impact and influence of computers on animation and imaging can be compared to the impact photography had on the visual arts of the late nineteenth century. Miniature painters and engravers feared that the new invention would replace them, and some of them even called it the "invention of the devil." Most new technologies that prove useful eventually become everyday technologies.

As mentioned earlier, computers have been used to create images since the 1950s, but the first artistic experiments with computer-based systems did not take place until the early 1960s. Most of the early animations and images produced with computers were not created in art studios but in research laboratories, and most of them used two-dimensional techniques at first. In fact, an unlikely partnership between Bell Labs physicist Billy Klüver and artist Robert Rauschenberg resulted in the 1967 *Experiments in Art and Technology* in New York City. But for the most part, the majority of individuals who created the early works of computer animation came from the

1.3.3 The 1988 short *Tin Toy* presents the story of a good-hearted toy who is willing to risk it all in order to save a baby in danger. (© Pixar Animation Studios.)

fields of science and engineering, and lacked formal training in the fine arts. Nevertheless, many of them displayed a strong artistic intention and a significant degree of aesthetic consciousness.

The computer systems that were used by these pioneers were not designed primarily for artistic creation. The IBM model 360 introduced in 1963, for example, was the first family of computers centered around a Fortran-based **time-sharing** system. Compared to today's computers, systems of those years were not very interactive, if they were interactive at all. The computer-human interface of the 1960s was typically opaque, cryptic, and not self-explanatory.

Using the early computer systems to create animations and images was not easy, so many of these early creators often had to put more effort into the process of creating the works than into the form and content of the works themselves. A few early computer artists were more concerned with the development of the computer-based imaging tools than with the style of their work. But in spite of all the limitations, the pioneers made effective use of the available technology.

The early computer-generated animations and images are the products of a technology still in the stage of development. The style of these early works was defined in a major way by the limitations of the computer equipment itself, and by the lack of computer programs that were capable of rendering complex images in a variety of ways. Very often complex methods and data structures did not yield correspondingly complex images. Among the American pioneers of computer art we can mention John Whitney, Sr., and Charles Csuri whose early computer-assisted animations date, respectively, from 1961 and 1966. (Csuri's later work can be seen in Figure 6.6.3.)

1.3.4 *Knickknack* presents the hilarious misfortunes of a Casanova toy snowman who, despite the fact that he charms the opposite toy-sex, can never seem to get close to them—his bad luck is always one step ahead of him. (© Pixar Animation Studios.)

One of the earliest experiments with computer-generated character animation was ***Mr. Computer Image ABC*** created in 1962 by Lee Harrison III with the **Scanimate** system at Computer Image Corporation (the system won an Emmy Award in 1972).

In 1968 the **Motion Picture Association of America (MPAA)** introduced a new rating system for movies that impacted the work done by filmmakers and animators. This rating system included four categories: G (General Audiences), M (Mature Audiences), R (Restricted), and X (No one under 17). A short time later the M rating was replaced by PG (Parental Guidance). Figure 1.3.11 presents today's updated listing of ratings across the major entertainment media.

Visual Milestones: 1970s

The panorama in computer art changed greatly during the 1970s because of the development of techniques for representing three-dimensional environments and because of the increased involvement of professional artists with computers. Computer-based animation and imaging systems became more interactive than what they were during the 1960s, but they were still not easy to use. Only a few of the artists who got interested in computer technology used it as their primary medium for artistic creation. In addition to their visual work, many of these early artists of three-dimensional computer animation and imaging also contributed to the technical development of their tools by collaborating in the development of software.

One of the most widely viewed works of three-dimensional com-

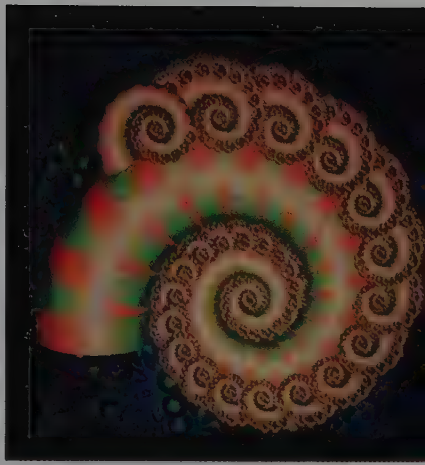
puter animation during the late 1970s was ***Voyager 2***, created by James Blinn and a team at the Jet Propulsion Laboratory (JPL) in California (Fig. 1.3.1). This work visualized the explorations of the Voyager 2 spaceship, and it is an excellent example of one of the earliest successful and extensive uses of image mapping techniques. Artist David Em, a visiting artist at JPL, created stills of fantastic planets with the same software used by scientists to render the planets of the Solar System.

Other notable examples of computer animation from this period include the 1974 animated film ***Hunger*** created by Peter Foldes under the auspices of the National Film Board of Canada. This work included striking computer-generated interpolations of key poses drawn by hand and painstakingly digitized into the computer software. ***Vol Libre***, a three-dimensional computer animation by Loren Carpenter, shows renderings of fractal mountains with great lyrical force (Fig. 1.3.2). ***The Joggler*** is an early example of a computer-animated human character attempting complex motion created at Information International Inc. (III). In 1974 The New York Institute of Technology (NYIT), in Old Westbury, New York, assembled a computer graphics research group with a notable roster of engineers and programmers. The goal was to develop computer graphics software and hardware to be used for commercial productions (Fig. 1.2.6). A few years later Industrial Light & Magic (ILM) was created to develop the visual effects for George Lucas' 1977 ***Star Wars***. This film brought visual effects to the foreground of mainstream culture, but the use of computer technology in this film was mostly limited to the computerized motion control systems used to move cameras and physical miniature models. The blue-screen compositing of the visual effects elements and plates in these films was achieved optically. In 1979 several members of the NYIT research group joined ILM with the goal of integrating computer graphics into visual effects production.

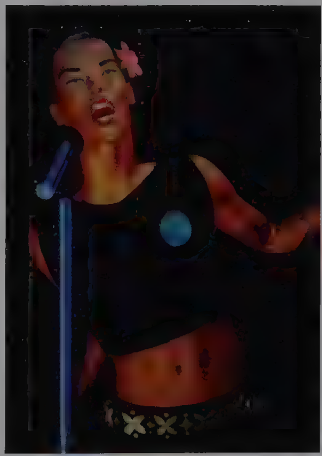
The commercial work done for advertising agencies at Digital Effects, III, MAGI, and Robert Abel and Associates is illustrative of computer animation in the late 1970s. These companies were active until the mid-1980s and then spawned other companies that continued their innovative spirit. Digital Effects was active from 1978 until the mid-1980s, III opened in 1974 and closed in 1982, MAGI was active between 1972 and 1987, and Robert Abel and Associates started in 1971 and closed in 1986.

Visual Milestones: 1980s

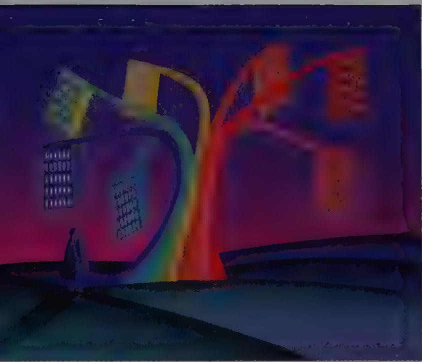
In the area of three-dimensional computer animation, the 1980s started with a few exceptional works and ended with a flurry of outstanding projects. This was due to many factors, such as the enhanced technology, the larger market, the maturing of the artists working in the field, and the entry into the workforce of the first art students who attended computer animation and imaging educational programs. The earliest realistic model of the full human figure from this decade is the virtual character Cindy created at III for the 1981



1.3.5 *Tendril* is among the first three-dimensional objects created with the recursive GROWTH algorithm. The colorful complex forms are derived from seed shapes like conch shells, tentacles, and coral. (© 1981 Yoichiro Kawaguchi.)



1.3.6 *Dozo* was the first female Synthespian™ performer created by Diana Walczak and Jeff Kleiser. She stars in their 1989 computer-generated film *Don't Touch Me*. (© 1989 Kleiser-Walczak Construction Co.)



1.3.7 Scene based on the designs of French illustrator Jean-Michel Folon, rendered to recreate the softness of pastel colors on paper. (Images courtesy of Toyo Links, from Tokyo Gas Company 1987 campaign.)

science fiction film *Looker*. The 1982 Disney film *TRON* was the first feature film with over 20 minutes of computer animation composited optically with live action; a few of the shots required dozens of passes with filters in front of the optical printer lens. *TRON* combined live action with three-dimensional computer animations created by the teams at Robert Abel and Associates, III, MAGI, and Digital Effects. For all of its visual innovation, however, this film was only moderately successful at the box office. The story behind this science-fiction film centered around a videogame designer who somehow ends up inside the virtual world of his creations and has to fight the game challenges that he himself created. The topic of this film also reflected the fact that the popularity of videogames hit a peak in the early 1980s, when Atari was the leading company in this arena. A decade earlier its founder, Nolan Bushnell, had developed the table tennis game *Pong* (1972) that helped launch the videogame industry.

In the area of visual effects for live action film, Industrial Light & Magic (ILM) continued the excellence in visual effects started just a few years earlier with George Lucas' *The Empire Strikes Back* (1980) and *Return of the Jedi* (1983). *Indiana Jones and the Temple of Doom* became in 1984 their first film to have an all-digital composite shot. *Young Sherlock Holmes* (1985) featured a somewhat convincing jointed character made of flat stained glass-like panels with texture mapping and transparency. In *Flight of the Navigator* (1986) keyframes of the live action footage were scanned and used for spherical reflection mapping to simulate interactive reflections as computer-generated objects travelled through the scene; three-dimensional morphing was also used in this movie. *The Abyss* (1989) revolved around the first three-dimensional computer animated character that was realistic enough to blend with the live action background plates. Because of the complex calculations of reflection and refraction, the team had enough time to render the different layers in each frame only once. The different passes (diffuse, specular, refraction, highlights) were composited optically, except for a single shot (when the safety door closes and the water creature is cut in half) that was composited digitally at 8 bits with Photoshop software.

1.3.8 *Locomotion* was one of the earliest (1988) three-dimensional computer animated shorts to use the technique of squash and stretch. (© PDI/DreamWorks.)

The **Genesis Effect** created in 1982 by ILM for the film *Star Trek II: The Wrath of Kahn* is also of historical interest because it was the first visual effects shot that was created entirely with three-dimensional computer animation techniques, the longest running sequence, and also because it is one of the earliest examples of procedural modeling and particle systems animation. *The Last Starfighter* was the first live action feature film to include a large amount of very realistic computer animation of highly detailed models. The basic production idea at the time was to replace the motion control cameras and the model photography with three-dimensional computer animation. The 28 minutes of computer animation for this 1985 film were animated and rendered with a Cray supercomputer at Digital Productions. To avoid aliasing artifacts most of the frames were computed at 20,000

lines of resolution and down-converted to about 1,000 lines.

Other notable works of the early and mid-1980s include ***Bio-Sensor*** created in 1984 at Osaka University and Toyo Links. This work is an impressive example of early figure locomotion and modeling with blobby surfaces (Fig. 1.2.5). The ***Brilliance*** commercial featuring a sexy female robot with convincing realistic motion was created by Abel and Associates, and also the first entirely computer-generated TV ad to be aired during a Super Bowl football game.

Also created during this period were the sublime simulations of light, fog, rain, and skies created at Hiroshima University; the intriguing non-edge simulations of clouds and smoke created by Geoffrey Gardner at Grumman Data Systems; and the first ray-traced imaging tests done by Turner Whitted at Bell Labs.

The mid-1980s also saw the rise of leading commercial production houses worldwide. In northern California, Pacific Data Images (PDI) was founded in 1980, Tippet Studios in 1983, and Pixar in 1985. In southern California, Boss Films was active from 1982 to 1997, and Digital Productions was in business from 1981 until the mid-1980s when it evolved into Whitney/Demos Productions for a few years. VIFX opened in 1984 and Rhythm & Hues Studios started in 1987. On the East Coast R/Greenberg Associates opened in New York City in 1981, the Kleiser-Walczak Construction Company was created in 1985, and Blue Sky Studios in 1987. Cranston-Csuri opened in 1981 in Columbus, Ohio, and closed in 1987, later spawning Metrolight. In 1982 Omnibus was started in Canada, and in 1986 it purchased both pioneer Robert Abel & Associates and Digital Productions before filing for bankruptcy a couple of years later. In Paris, Buff opened in 1985, Mac Guff Ligne in 1986, and Sogitec was active from 1986 to 1989 when it merged with TDI to create Ex Machina. CA Scanline opened in Münich in 1989. In Japan Toyo Links opened in 1982, the Japan Computer Graphics Lab (JCGL) was active from 1981 until it was purchased in 1988 by the videogame company Namco, and Polygon Pictures opened in 1983.

Throughout the 1980s two constants exemplify the excellence reached by three-dimensional computer animation during the decade. On one hand there was the engaging and amusing character animations by the animation team at Pixar led by John Lasseter, including ***Luxo Jr.*** (1985), ***Red's Dream*** (1987), ***Tin Toy*** (1988), and ***Knickknack*** (1989). These Pixar projects not only pushed the RenderMan shading language to its limits, but also proved that the traditional principles of character animation could be applied to computer-generated works (Figs. 1.1.1 and 1.3.3–1.3.4). On the other hand there was ***Growth***, a series of semi-abstract animations by Japanese artist-programmer Yoichiro Kawaguchi. The series portrays imaginary underwater creatures generated with procedural techniques (page 1 and Fig. 1.3.5).

The late 1980s witnessed experimentation with a wide variety of techniques ranging from the simulation of natural-looking hair growth to rigid body dynamics and modeling fabric with visible threads.

1.3.9 The character Lotta Desire from *The Little Death* had a single and continuous skin that covered her entire body, including eyes with animateable irises. This character was also one of the first to use an early version of the surface subdivision technique to raise the polygonal resolution for film output, about 5,000 polygons that were subdivided to 20,000 at rendertime. Lotta Desire also contained internal displacements, not morph targets, to achieve her facial expressions. (© 1989 Matt Elson.)

1.3.10 The hilarious *Technological Threat*, in 1988, combined three-dimensional wireframe computer animation of the environment and boss with traditional hand-drawn animation of the employees. (© 1999 Kroyer Films, Inc.)

Rating Systems Across Entertainment Media	
Movies, MPAA	
G	General Audiences
PG	Parental Guidance
PG-13	Parents Strongly Cautioned
R	Restricted
NC-17	No one 17 and Under
Television, TV Parental Guides	
TV-Y	All Children
TV-Y7	Older Children (7+)
TV-Y7-FV	Older Children (7+), Fantasy Violence
TV-G	General Audience
TV-PG	Parental Guidance Suggested
TV-14	Parents Strongly Cautioned (14+)
TV-MA	Mature Audience Only (17+)
Video/Computer Games, ESRB	
eC	Early Childhood (3+)
E	Everyone (6+)
E10+	Everyone (10+)
T	Teen (13+)
M	Mature (17+)
AO	Adults Only (18+)
RP	Rating Pending

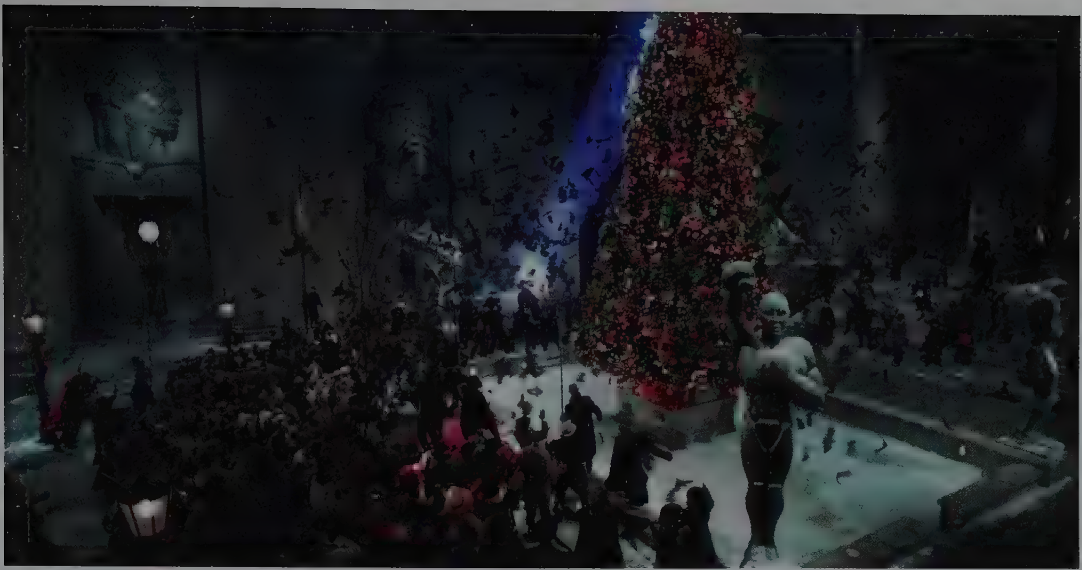
1.3.11 Ratings across media by the Motion Picture Association of America, the TV Parental Guidelines Monitoring Board, and the Entertainment Software Ratings Board.

Stanley and Stella: Breaking the Ice, produced by Symbolics Graphics and Whitney Demo Productions in 1987, is a solid and amusing early example of flock animation. ***Don't Touch Me,*** created in 1989 by the Kleiser-Walczak Construction Company, represents one of the earliest tours de force in character animation with motion capture techniques (Fig. 1.3.6). The female singer in this piece displayed more body animation and faster motion than any previous attempt; the animation was achieved by applying the motion of a live singer to the virtual character. The demo reels of design and production studios such as Rhythm & Hues and Metrolight in California, Ex Machina in Paris, Digital Pictures in London, and Toyo Links in Tokyo (Fig. 1.3.7) are representative of the commercial work of the period.

Independent short computer animations created during the late 1980s have many inspired examples. ***Burning Love*** created by Pacific Data Images in 1988 displayed an emotional quality that had not been seen in too many computer animations of the period, and it also used one of the earliest painterly treatments of three-dimensional computer rendering. ***Locomotion,*** also by Pacific Data Images (1988), illustrated the story of a charming train engine that overcomes a broken bridge and, in the process, displayed great understanding of the traditional animation principles of squash and stretch (Fig. 1.3.8). ***The Little Death,*** created by Matt Elson at Symbolics Graphics in 1989, applied the technique of displacement animation onto detailed models of the human figure (Fig. 1.3.9). ***Grinning Evil Death*** (McKenna and Sabiston at MIT's Media Lab) had a sassy, almost sinister sense of humor that was uncommon in computer animations of this period. ***Technological Threat,*** created by William Kroyer in 1988, combines hand-drawn animation with three-dimensional computer wireframe animation to present a hilarious view of the automated office (Fig. 1.3.10).

During the second half of this decade Walt Disney Feature Animation, one of the dominant forces in traditional animation, began to experiment with three-dimensional computer animation in its animated feature films (Fig. 1.1.3). ***The Black Cauldron,*** released in 1985, was the first Disney animated feature film that used computer graphics technology in a small section of the movie to simulate a flying visible light source. ***The Great Mouse Detective*** (1986) contains a one-minute chase sequence almost at the end of the movie where the hero tries to rescue the heroine from the villain in a landscape of menacing gears that threaten to crush them as they try to evade the villain. The gears were modeled and animated with three-dimensional computer animation techniques, and then output as drawings on paper with a pen plotter. This allowed them to be integrated into the traditional production process of the time.

Many of the cityscapes in ***Oliver & Company,*** a 1988 Disney release, are populated with animated three-dimensional cars. In addition, some of the car interiors are settings for conversations between hand-drawn cartoon characters, and some close-ups of car exteriors are also backdrops for shots with the canine characters that drive the film. ***The Little Mermaid*** (1989) was the last Disney animated feature



1.4.1 Animating flocks of bats in the film *Batman Returns*. The behavior of the bats is based on a flock animation computer model originally developed by Craig Reynolds in 1987. (Additional software written by Andy Kopra. *Batman Returns*™, © 1992 DC Comics. Image courtesy of VIFX.)

film to use traditional ink and paint production techniques. The very last scene in the film, where the crowds wave good-bye, was done digitally with Disney's proprietary **CAPS** software (Computer Animation Production System).

In the area of live action films, *The Abyss* is a 1989 feature film that crowns the decade with convincing examples of computer-generated visual effects, seamless compositing with 70 mm live action footage, and a complex production created to a great extent with off-the-shelf systems. One of the most striking moments in this film takes place when the computer-animated water creature emulates the facial expressions of the human actor who also touches the virtual creature with her hand. The first Game Developers Conference took place in 1987, and the MPAA expanded its rating system in 1984 to include a PG-13 category (Fig. 1.3.11).

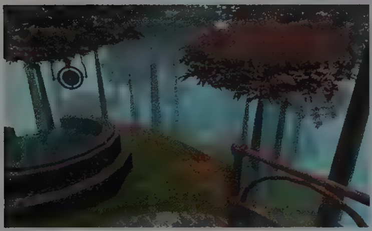
1.4 Visual Milestones: 1990–1999

Few suspected the explosion in popularity, artistic creativity, and significant revenue that computer animation and visual effects were going to play during this decade, including the beginning of a renaissance in visual effects and the creation of *Toy Story*—the first all-three-dimensional computer animation feature.

Visual Milestones: Early 1990s

The early 1990s were characterized by refined examples of computer animation as well as a successful revival of special effects for feature films. Three-dimensional computer animation and imaging during this period became quite complex and full of varied styles and

1.4.2 The facial animation of this surprised virtual Marilyn Monroe was generated using the SMILE system, an early 1990s multilayer animation system with muscle deformation at the lower level and a high-level language to specify emotions, speech, and eye motions. (© 1991 Nadia Magnenat-Thalmann, MIRALab, University of Geneva, and Daniel Thalmann, Computer Graphics Lab, EPFL, Lausanne.)



1.4.3 *Myst* pushed the limits of realistic rendering for what was possible with 8-bit color CD-ROM computer games in the early 1990s. (Screen shot from Myst® CD-ROM computer game. Game and screen shot © 1993 Cyan, Inc®. All rights reserved.)

attitudes. Many of the projects from this period encompass an exciting body of work and a variety of styles and techniques. By the middle of the decade three-dimensional computer animation and imaging had become a mature and specialized field that finally gained a fair amount of wide recognition. Computer animators and digital artists in general were in great demand due to the increased production slate of visual effects films, animated films and television series, and computer and platform games.

The early 1990s witnessed a fair amount of company transitions in the field of commercial computer animation and visual effects. The Mill in London and Santa Barbara Studios started in 1990, and Digital Domain opened in 1993. The same year Square opened with the original goal of creating animation for both games and feature film. Sony was the first Hollywood movie studio to consider developing an in-house visual effects and computer animation facility. The result of this idea was Sony Imageworks, which opened in 1992. Later in the decade other Hollywood studios purchased independent computer animation and/or visual effects production houses. Another notable business event of this period was the agreement between Disney and Pixar to codevelop, produce, and distribute several animated feature films.

Several feature films created during the early 1990s used computer-generated visual effects extensively. ***Terminator II***, for example, is a landmark 1991 film by James Cameron with computer animation by Industrial Light & Magic. This film was the first mainstream blockbuster movie to include outstanding three-dimensional morphing effects, the first convincing simulation of natural human motion,

global reflections and even a few self-reflections when the digital actor walks through the metal bars. *The Lawnmower Man*, a 1992 film with computer animation by Angel Studios, was the first feature film of the decade that explored the topic of virtual reality with computer animation. *Batman Returns* is a 1992 stylized production with effective examples of flock animation (Fig. 1.4.1). The same year *Death Becomes Her* used extensive digital retouching by removing the head of an actress and later tracking a shot of a talking real head onto the body. *Jurassic Park*, a 1993 film by Steven Spielberg with computer animation by ILM, is an early example of using inverse kinematics skeletons, skin, and local deformation for each muscle. This movie was also the first example of a computer-generated human stunt double, and a great example of hyperrealistic rendering. A gigantic amount of processing was performed in a relatively short period of time. For the first time in a live action feature film, digital compositing replaced almost entirely photochemical optical compositing for the perfect integration of live action and animatronics (for the close-ups of dinosaur heads) with computer-generated images. *The Flintstones* (1994) is an early example of fur rendering, used for the saber tooth tiger.

Other notable examples of computer animation during the early 1990s include *Primordial Dance* (1991) and *Liquid Selves* (1992), both beautiful examples of computer animations created by Karl Sims with particle systems techniques; the irreverent and amusing *Le Xons Crac-Crac* and *Baston* created in 1991 by Ex-Nihilo and Mac Guff Ligne; William Latham's hypnotic *Mutations* (1991); *Virtual Marilyn* (1991) created with an early muscle-based facial animation system by Nadia Magnenat-Thalmann and Daniel Thalmann (Fig. 1.4.2); *Don Quichotte,* an ambitious keyframe animation created in 1991 by Video System; and *Leaf Magic* (1991), a good example of animation with motion dynamics by Alan Norton.

Some of the outstanding work for television commercials includes the 1993 *Coca-Cola Polar Bears* by Rhythm & Hues Studios (Fig. 1.4.4) and the 1994 *Listerine Arrows* by Pixar. Many ambitious and exquisite architectural visualizations were also created in the early 1990s, including *The Seven Wonders of the World* (1992) by Electric Images in England, *The Ancient World Revisited I-III* (1990–94) by Taisei Corp., and *De Karnak a Louqsor* (1992) by Ex Machina. In 1993 *Myst* set the standard for prerendered three-dimensional computer animation for computer games (Fig. 1.4.3). The early 1990s also saw great examples of broadcast-quality computer animation created entirely with off-the-shelf microcomputer systems. *Babylon 5*, for example, is a 1993 TV series and the first mainstream example of high-end three-dimensional computer animation that was initially produced entirely on 32-bit Amiga and Macintosh microcomputers. The now ubiquitous Music Television Channel (MTV) went on air in 1981 and started its Music Video Awards program in 1984, but it wasn't until the 1990s that **music videos** came of age as a viable and original medium for cutting-edge

1.4.4 The first polar bear commercial created by Rhythm & Hues Studios was created in 1993, above. Middle and below are frames from the 1996 and 1998 editions of the commercial. (© The Coca-Cola Company. "Coca-Cola," the Coca-Cola Polar Bear design, and the "Coca-Cola" Contour Bottle are trademarks of The Coca-Cola Company.)

1.4.5 *Moxy*, the virtual host of Cartoon Network's first original animated program, was animated with a live motion capture and motion control system. (© 1993 Cartoon Network, Inc. See Figure 12.2.6 for full credit.)

1.4.6 *Megasónicos*, also known as *Megasonikoak*, was the first 3D computer animation feature to be produced in Europe. (© 1997 BALEUKO, S.L.)

computer animation and visual effects (Figs. 1.4.14 and 1.4.15).

Motion capture systems for character animation experienced an intense development during this period. Some of these efforts included the Facetracker system developed by SimmGraphics to animate the Super Mario character; ***Moxy,*** a virtual TV host animated by Colossal Pictures for the Cartoon Network (Figs. 1.4.5 and 12.2.5); Acclaim Entertainment's optical system with up to 70 sensors for simultaneous two-person capture (Fig. 12.2.5); and a variety of commercially available motion capture hardware and software.

During the early 1990s computer animation at Walt Disney Feature Animation went from a mere novelty to a significant standard component in the digital production process. Disney's 1990 release ***The Rescuers Down Under*** was the first Disney animated feature film to be produced entirely with the first version of the CAPS software. This landmark event ended 53 years (since 1937) of inking and painting acetate cels. This film also contained a moving vehicle and a few props created with three-dimensional computer animation. ***Beauty and the Beast*** (1991) includes the memorable ballroom scene where the animated camera follows Beauty and Beast as they dance in a three-dimensional environment that included columns with marble textures and a detailed chandelier. This film was also the first animated feature film to be nominated for an Academy of Motion Picture Arts and Sciences award in the Best Picture category.

In ***Aladdin***'s magic carpet we see the first example of Disney character-driven computer animation. The magic carpet's organic surfaces and detailed texture were animated with cartoony squash and stretch and impeccable timing. The complex textures mapped on the carpet would have surely been a challenge to animate, ink, and paint with traditional production techniques. In addition the three-dimensional computer animation in this 1992 movie also included the tiger-cave head rising from the sand dunes, the lava sequence, and countless other animation effects. The computer-generated stampede of wildebeests in ***The Lion King*** (1994) is one of the most striking moments in modern feature animation. It sets up with dramatic force and visual power the terrible events that will take place before the sequence is over. The wildebeests were animated with a crowd simulation system that predated many of the crowd systems that became popular later in the decade. The wildebeests' shadows and dust were also created with a three-dimensional computer animation system, as was the fire effects animation.

Visual Milestones: Late 1990s

The second half of the 1990s exploded with productions rich in three-dimensional computer animation in the fields of live action films with special effects, animated feature films, and computer or platform games. In 1995, Pixar's ***Toy Story*** became the first feature animated film to be entirely created with three-dimensional computer animation techniques (Fig. 12.1.1). Three years later two other pro-

jects joined the competitive world of three-dimensional computer animation: DreamWorks' *ANTZ* (Figs. 2.7.10 and 2.7.11), and Pixar's *A Bug's Life*. Three other animated feature films—each from a different studio—were released with wide distribution in the United States during 1998: Disney's *Mulan*, Nickelodeon's *Rugrats*, and DreamWorks' *Prince of Egypt*. The animated feature *Princess Mononoke* by Hayao Miyazaki was released in Japan during 1997. Not only did this film present a magical view of the world, including a hovering multi-tentacled monster created with 3D computer animation and non-realistic rendering, it also set box-office and TV records. At the time this film was the highest grossing of any film in Japan, and it scored the eighth best TV viewership rating ever. *Los Megasónicos*, also known as *Megasonikoak* in Basque, made history in 1997 as the first European three-dimensional computer animation feature (Fig. 1.4.6).

1.4.7 The flea from 1995's *La cité des enfants perdus*. (© Claudie Ossard. Directors: Jeunet & Caro. Visual effects: Buff.)

Not only was the volume of animation production during 1998 impressive, but also the quality and creative diversity was unparalleled in the history of animation. The 1999 animated feature film releases were equally impressive: Disney's *Tarzan* and *Fantasia 2000*, Pixar's *Toy Story 2*, Warner Bros.' *Iron Giant*, and Paramount's *South Park*. *Fantasia 2000* became the first animated feature film (over 90 minutes long) to be released exclusively for the IMAX large-screen format for the initial four-month run. *Toy Story 2* increased the visual complexity of the computer-generated environments and the human characters in particular; *Iron Giant* presented an unlikely hero that was rendered with a non-photorealistic RenderMan shader.

The activity and quality of the craft in the field of visual effects during the later part of the decade was also very high. Just consider some of the many films that featured fully computer-animated main characters with live actors: *The Lost World: Jurassic Park* (1993), *Jumanji* and *Species* (1995), *Dragonheart* (1996), *Titanic*, *Starship Troopers*, and *Mars Attacks!* (1997), *Mighty Joe Young*, *Mouse Hunt*, and *Godzilla* (1998), *The Mummy* and *The Phantom Menace* (1999). The decade—and the millennium—in visual effects was capped with four popular effects-oriented films: the stylish *The Matrix*, winner of the 1999 AMPAS award for Best Visual effects, *Stuart Little* with its innovative combination of cartoon and realistic action (Figs. 14.3.3 and 14.3.4), the sleeper hit *The Mummy*, and the much-heralded Star Wars prequel *The Phantom Menace*. The visual effects for the former movie were done at Mass Illusion, the later two movies were done at Industrial Light & Magic. *La cité des enfants perdus* (The City of Lost Children) was a 1995 French movie with ground-breaking visual effects and ray tracing rendering done with Mental Ray software (Fig. 1.4.7). The same year *Casino* used Lightscape radiosity software to create realistic set extensions, *Waterworld* incorporated water sim-

1.4.8 Some of the main characters in Chris Landreth's *Bingo*, one of the earliest computer animations rendered in a realistic style, based on improvisational theater techniques. (© Alias|Wavefront, a division of Silicon Graphics Limited.)

1.4.9 The colors in this watercolor simulation are animated to reflect the changing time in day as well as the mood of the Fisherman. (*Fishing*, a 1999 PDI short film by David Gainey.)

1.4.10 The character in *Alien Song* meets an unexpected end while disco-dancing to Gloria Gaynor's *I Will Survive*. See Figure 8.4.4 for a related image. (© 1999 Victor Navone.)

ulated with Areté software (Fig. 5.5.10), and *Jumanji* rendered fur on a scale larger than ever before. ***Babe*** advanced the techniques for removing, tracking, and replacing talking animal heads. During 1996 the computer-generated creature in ***Dragonheart*** was the co-star of the movie, RenderMan shaders were used in ***Mission Impossible*** to match the look of the film stock used, and a massive dynamics simulation was used to create tornados for ***Twister***. During 1997 award-winner ***Titanic*** used computer-generated water, massive digital compositing, and motion capture to create digital extras on the deck of the ship; ***Spawn*** achieved a dramatic effect in the cloth animation by combining realistic rendering with exaggerated keyframe animation; and ***Mars Attacks!*** included amusing slapstick comedy and extensive cloth animation. In 1998 ***What Dreams May Come*** perfected camera tracking techniques to facilitate the *animated painting* look developed for the movie, and ***Mighty Joe Young*** perfected hair rendering techniques. The 1999 award-winner ***The Matrix*** used image-based rendering as well as the *frozen time* visual effect (see Chapter 13); ***Star Wars: Episode One*** had memorable spacecraft racing and battle scenes, with some of the best crowd simulation work done until then; ***Stuart Little*** brought new life to the live action movie with a computer-generated hyperrealistic cartoon character with simulated wet cloth and wet fur; and ***Fight Club*** used image-based modeling and rendering (before ***The Matrix***) with several impossible camera moves through walls.

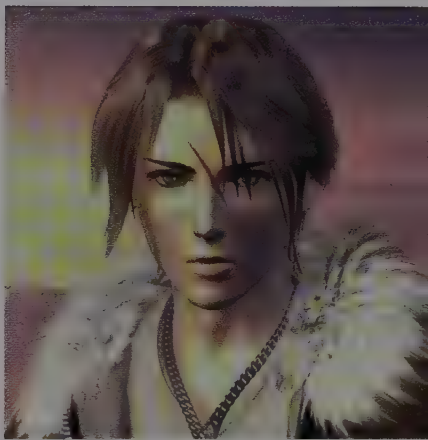
When compared to films of previous decades the visual effects films of the late 1990s had more and more complex digital and traditional effects shots. ***Terminator II***, for example—a landmark 1991 production that redefined the use of computer animation in live action movies—had approximately 150 visual effects shots including 44 digital effects. In 1995 ***Batman Forever*** had about 250 visual effects shots. In 1997 ***Titanic*** had close to 550. In 1998 ***Armageddon*** had around 240, and ***Godzilla*** had close to 400. In 2000 ***How the Grinch Stole Christmas*** had about 600 visual effects shots with about 300 of them computer-generated, including falling and melting snow; and ***The Perfect Storm*** had around 225 shots with computer-generated virtual stunt actors, 40 of them, and 33 minutes of computer-generated waves.

Production of computer animation and visual effects for feature film during the late 1990s was mostly dominated by the companies that had established themselves as leaders during the previous ten years. The major newcomers include Foundation Imaging (1992), Blur Studios, Banned from the Ranch (both opened in 1995), and Centropolis (1996). Continuing the trend set by Sony earlier in the decade with the creation of its internal visual effects and computer animation facility, in 1996 DreamWorks purchased an interest in Pacific Data Images, Fox purchased VIFX, and Disney purchased industry pioneer Dream Quest Images. A year later Fox purchased Blue Sky Studios. Warner Digital was active from 1995 to 1997.

Some of the most polished short computer animation works

that we have ever seen were also produced during this period. This includes *Geri's Game* by Jan Pinkava at Pixar in 1997 (Fig. 1.2.9), *Bunny* by Chris Wedge at Blue Sky Studios (Figs. 7.2.4 and 10.1.1) and *Bingo* by Chris Landreth at Alias Research in 1998 (Figs. 1.4.8 and 4.2.7), and *Tightrope* by Daniel Robichaud at Digital Domain in 1999 (page 293 and Fig. 12.5.8). Not only were these shorts produced by different groups, but each had a recognizable style that made it unique. Both *Geri's Game* and *Bunny* won AMPAS awards in the Best Animated Short category. The original non-photorealistic *Fishing* by David Gainey and the simple walk and character study *Alien Song* by Victor Navone were both released in 1999 (Figs. 1.4.9 and 1.4.10).

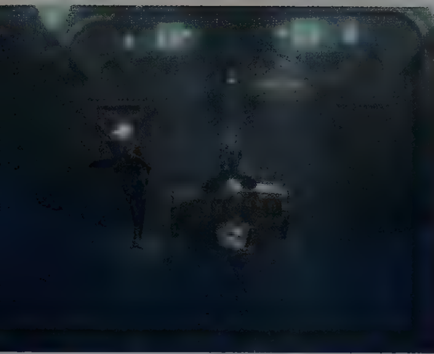
The games industry, fueled by the new game platforms, grew tremendously. In 1998, for example, Nintendo released *The Legend of Zelda: Ocarina of Time* for the N64. Between its launch date of November 23 and the end of the year, Nintendo reported $150 million in sales of 2.5 million copies. As a point of reference, one of the higher-grossing movies during that year was Disney/Pixar's *A Bug's Life*, which collected $114 million at the box office. A few games from this period with innovative character computer animation include *Soulblade*, my personal favorite fighting game from those years, Nintendo's *Super Mario 64*, adventure and role-playing *Ultima Online*, strategy and wargames *Age of Empires* and *Civilization*, sports games *NHL 98* and *NBA Live 97*, action games *Duke Nukem 3D* and *Tomb Raider*, the arcade version *Virtua Fighter 3,* and a few installments in the *Final Fantasy* series by SquareSoft (Fig. 1.4.11).



1.4.11 Two of the main characters from the *Final Fantasy* videogame. (© 1998 Square Co., Ltd. All rights reserved.)

Some of the notable three-dimensional computer animation for early **location-based entertainment**—or **LBE rides** as they are known—include *The Volcano Mine Ride* (1995), and *Seafari* and *Race for Atlantis in Imax 3D* created by Rhythm & Hues Studios in 1994 and 1998, respectively (Figs. 1.4.12 and 1.4.13). Arcade games got away from shooting and fighting, and experienced growth in the areas of sports simulations like skiing, snowboarding, and jet skiing.

Peter Gabriel's polished music video *Kiss that Frog* won Best Special Effects in the 1994 MTV Video Music Awards (Fig. 1.4.14). A couple of the more memorable commercials included the *Dance Fever* commercials of dancing cars created by R/Greenberg Associates for Shell Oil in 1995, and *Virtual Andre* created from a motion-captured Andre Agassi by Digital Domain in 1997. There



1.4.12 The future of location-based entertainment is being defined by computer-generated rides like *Seafari*, a hyperrealistic computer-generated motion ride that takes the audience on an underwater rescue mission. (© 1994 MCA/Universal. Courtesy of Rhythm & Hues Studios.)

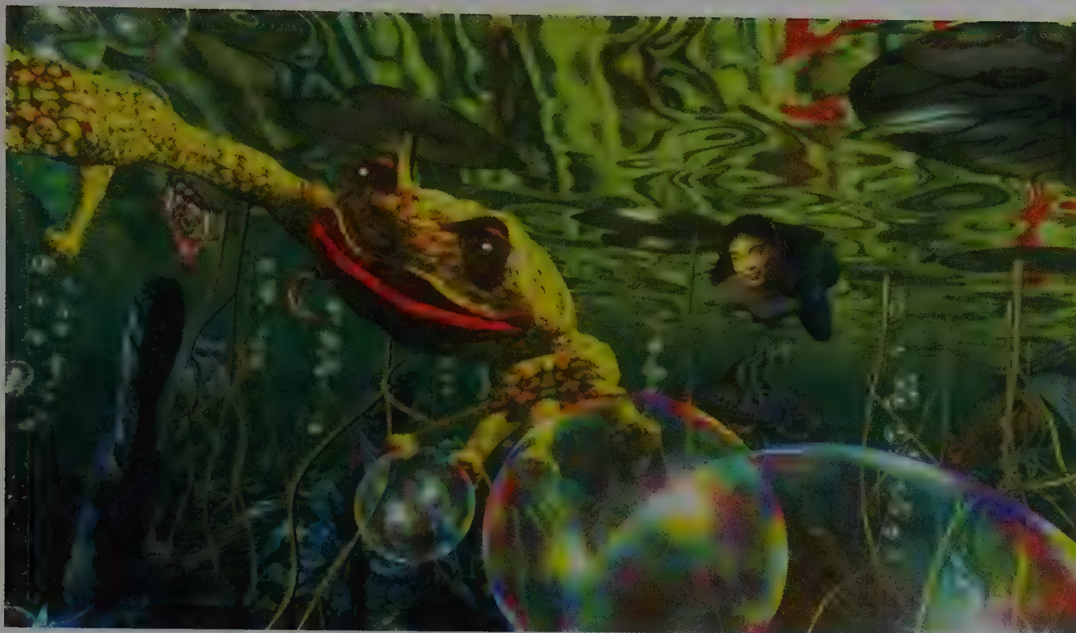
1.4.13 View of *Race for Atlantis* projected on the curved surface where the ride takes place. (Courtesy of Forum Ride Associates.)

was also a lot of activity in the area of animated television series, with new all-computer-animated series like ***Beast Wars*** and ***Reboot*** by Mainframe Entertainment (pages 58 and 449), ***Rednecks*** by Foundation Imaging, and charming ***Rolie Polie Olie*** (premiered in 1998) by Nelvana and Paris-based Sparx. Likewise, feature-length direct-to-video releases grew significantly during the decade.

Throughout the late 1990s traditional animation production at Walt Disney Feature Animation moved closer to three-dimensional computer animation, resulting in several examples of blending tradition with innovation. The rough animation for the canoe that ***Pocahontas*** (1995) rides down the river, for example, was created with three-dimensional computer animation techniques and match-moved to other elements in the scene. The face of the wise Mother Willow character was also animated with three-dimensional computer animation. Disney's 1996 production of ***The Hunchback of Notre Dame*** showcases three-dimensional computer-generated confetti and crowds, motion blur effects, and a few props. In the final shot of the "Sanctuary!" sequence, where Quasimodo rescues Esmeralda from being burned at the stake and carries her to the top of Notre Dame, the three-dimensional model of the rosetta and its architectural details add dimensionality to the shot.

The 1997 release ***Hercules*** includes a memorable segment where the hero fights a multiple-headed Hydra monster. As he cuts some of the heads others immediately pop up in a menacing and fierce way. In addition a morphing technique was used in the opening shots of the movie to blend still paintings of clouds, columns, and other background paintings. ***Mulan*** (1998) displays a large variety of three-dimensional computer-animated props that blend seamlessly with the hand-drawn elements and hand-painted backgrounds. These props include flags, arrows, and carts. There is also a fair amount of effects animation like the smoke and fire from the flaming arrows and, of course, the Hun charge, which is somewhat reminiscent of moments of the wildebeest sequence in the 1994 blockbuster ***The Lion King***. The Hun charge sequence, however, is enhanced with a couple of low-flying traveling shots that add drama to the danger and uncertainty of the sequence. Hand-drawn two-dimensional characters as well as some background elements were applied to three-dimensional billboards in order to populate scenes in this movie. The 1999 production of ***Tarzan*** offered lush jungle environments created with Disney's proprietary software Deep Canvas to recreate 2D brushstrokes on three-dimensional geometry. Many of the procedural effects animation (including water) also add to the story. From the beginning the movie was produced keeping in mind that it would be released in theaters both as a film print and as a pioneer in the digital cinema, or D-cinema, projection technology. For that reason all the final frames in the film are 100 percent digital, including the 540 feet of rolling animated credits at the end of the movie.

A couple of refinements and additions to the ratings systems took place during this decade. In 1990 the MPAA renamed their X



1.4.14 *Kiss that Frog* is a rock video based on the music of Peter Gabriel. A rich array of shading techniques was used to portray exotic creatures in an environment that also contains live action characters. (© 1993 MEGA/ Real World. All rights reserved. Courtesy of Angel Studios, Carlsbad, California.)

rating to NC-17. In 1994 the Entertainment Software Ratings Board issued a ratings system for computer games, and a new system for rating television programs based on the existing MPAA categories was put in place in 1997 (Fig. 1.3.11).

1.5 Visual Milestones: 2000–Today

This decade witnessed a coming of age of computer animated features, where the initial fascination and novelty gave way to works that pushed the envelope on the technical, creative, and entertainment fronts. Visual effects movies also continued to become more ambitious and sophisticated: the number and complexity of VFX shots in effects-driven movies continued to increase, and incorporating high-quality VFX shots in non-effects movies became prevalent.

Visual Milestones: Early 2000s

The early years of this decade witnessed growth in the quantity of movies incorporating visual effects as well as the total number and quality of effects shots per show. The statistics of ***Star Wars: Episode II***, for example, are impressive: 2,200 visual effects shots, 10,200 visual effects elements, 5 million frames, 929 animated shots, 20 different cuts of the movie, and a crew of over 250 digital artists who completed every day of the production the equivalent of one worker-year of work. The choreography of the camera and the action is one of the most complex ever produced for a live action effects production. George Lucas, the director, also pushed to its limits the high-

MTV's Video Music Award: Best Special Effects

1984 *Rockit*, Herbie Hancock
1985 *Don't Come Around Here No More*, Tom Petty
1986 *Take on Me*, a-Ha
1987 *Sledgehammer*, P. Gabriel
1988 *Hourglass*, Squeeze
1989 *Leave Me Alone*, Michael Jackson
1990 *Sowing the Seeds of Love*, Tears for Fears
1991 *Falling To Pieces*, Faith No More
1992 *Even Better Than the Real Thing*, U2
1993 *Steam*, Peter Gabriel
1994 *Kiss that Frog*, P. Gabriel
1995 *Love Is Strong*, Rolling Stones
1996 *Tonight, Tonight*, Smashing Pumpkins
1997 *Virtual Insanity*, Jamiroquai
1998 *Frozen*, Madonna
1999 *Special*, Garbage
2000 *All Is Full of Love*, Björk
2001 *Rock DJ*, Robbie Williams
2002 *Fell in Love with a Girl*, The White Stripes
2003 *Go With the Flow*, Queens of the Stone Age
2004 *Hey Ya!*, OutKast
2005 *Speed of Sound*, Coldplay
2006 *We Run This*, Missy Elliott
2008 *Good Life*, Kanye West with T-Pain

1.4.15 List of music videos that have won the Best Special Effects category in MTV's Video Music Awards.

definition (HD) and blue screen technologies available in 2002, proving that a feature movie can be made entirely with virtual sets, extensive previsualization, and compositing techniques. ***Spy Kids 2*** was another movie that followed with great results a smaller scale of the same production pipeline: shoot on HD taking advantage of green-screen techniques, digitize the live action, create computer-generated visual effects, composite all elements using the digital intermediate process, and output to desired media (Figs. 2.7.3 and 13.1.1).

The three installments of ***The Lord of the Rings*** (2001–2003) showcased stylized environments and creatures created with myriad effects techniques, ranging from in-camera effects to computer animation. Of special note are the eerie and emotionally convincing ***Gollum*** character (Fig. 12.2.7), the crowd animation system (Fig. 12.6.1), the superb color timing, and the fact that elements for the three episodes of this saga were shot simultaneously. The 2001 movie ***Pearl Harbor*** used massive amounts of computer-generated set extensions and props to replicate many of the ships and most of the airplanes. Billions of particles were also used to simulate the smoke and fire of bomb explosions. An effective rigid-body dynamics system was developed to make the crafts' explosions more real. Ang Lee's ***Hulk*** (2003) offered a unique combination of cartoon motion and realistic motion. The sequels ***Matrix: Reloaded*** and ***The Matrix: Revolutions*** rounded up the innovative style of the original.

Because of the increased availability of high-quality visual effects this was also a period that saw a rise in the number of feature movies with excellent and subtle supporting visual effects that do not "carry the picture" but provide important accents to the storytelling. A few examples include the rain of frogs in ***Magnolia*** and the rose petals in ***American Beauty*** (1999); the boulder plunging into the swimming pool in ***Sexy Beast*** (2000); ***Amélie***'s visible heart (2001); and in 2002 the dream sequences in ***Frida***; Lechter removing the scalp in ***Hannibal***; and Adrien Brody's head digitally composited onto the body of a piano virtuoso in Polanski's ***The Pianist***.

Final Fantasy marked the first attempt to create an entire animated feature movie with motion capture techniques (Figs. 5.5.13 and 9.2.4), and in spite of stunning visuals the commercial results were mixed. Warner Bros.' ***Polar Express*** (2004) was another early animated feature based on extensive motion capture. ***Synchronicity*** was one of the first independent shorts to use motion capture techniques (Fig. 1.5.1). A pair of movies featuring ***Barbie*** also made use of motion capture techniques (Figs. 1.5.2 and 12.2.8). Five years in the making, Disney's ***Dinosaur*** blended live action backgrounds with computer-generated characters. Fox's ***Ice Age*** presented a mixture of slapstick comedy and drama in a beautifully rendered saga (page v and Fig. 10.4.10). DreamWorks' ***Spirit*** made innovative use of three-dimensional computer animation techniques and "cartoon-style" non-photo-realistic rendering. Disney's ***Atlantis*** and ***Treasure Planet*** display spectacular integration of 2D and 3D animation techniques, but the latter failed to capture the interest of audiences during its initial release.

Partly because of the growth in quality and quantity in animation production the American Academy of Motion Picture Arts and Sciences (AMPAS) added a new category in 2001 for Best Animated Feature Movie in their yearly competition. DreamWorks' *Shrek*, with its mixture of irreverent humor and unique stylized rendering, was the winner of the first award in this category (Figs. 2.2.6 and 12.4.1). Pixar's *Monsters Inc.* featured a green one-eyed monster, a blue-haired monster with 3 million hairs (Figs. 2.2.6 and 5.5.13), and a lovable girl, and *The Incredibles* featured a super-powered family animated with significant squash and stretch. DNA's *Jimmy Neutron Boy Genius* proved that medium-size companies could deliver good-quality three-dimensional animation with off-the-shelf software, NewTek's Lightwave, and within reasonable budgets (Fig. 1.5.4). Japanese animation master Hayao Miyazaki's *Spirited Away* was the 2002 winner in this category, Pixar's *Finding Nemo* and *The Incredibles* won in 2003 and 2004 (see the timetables at the end of this chapter for a listing of all nominated movies). A few years earlier the SIGGRAPH Conference also created the Best Animated Short and Jury Honors awards, which were won in 1999 by Chris Wedge's *Bunny* and Piotr Karwas' *Masks*, respectively (Figs. 10.1.1 and 8.4.3); in 2000 by the cinematic opening in the *Onimusha* PS2 videogame (Fig. 11.3.6) and *Stationen* by Christian Swade-Meyer; and in 2004 by Sejong Park's *Birthday Boy* (Figs. 4.5.5 and 7.2.9) and Chris Landreth's Academy Award winner *Ryan* (Fig 10.3.4). Tomek Baginski's *The Cathedral* (Figs. 8.3.7 and 8.6.5) and Sam Chen's *Eternal Gaze* (Figs. 10.4.2 and 12.5.10) were the winners of the Best Animated Short in 2002 and 2003, respectively. In 2002 the Visual Effects Society (www.ves.org) instituted an awards program to recognize the different visual effects specialties (Figs. 1.5.3 and 13.12.3).

Computer animation and visual effects production was active and grew internationally during this period. Paris-based Duran completed memorable supporting effects for *Amélie* and a number of music videos including *It's Not the End of the World* by the Super Furry Animals, and embarked in the production of *Immortel (ad vitam)*, an all-computer-animated feature based on the graphic novel *La femme piège* by comic book artist Enki Bilal (Fig 4.2.6). *Kaena, la Prophétie* is another French computer-animated feature completed in 2003 (Fig. 8.3.11), as well as the Danish irreverent comedy *Terkel in Trouble* (Fig. 9.4.6). Yimou Zhang's *House of Flying Daggers*, Stephen Chow's *Kung Fu Hustle*, and Timur Bekmambetov's *Night Watch* are examples of the increasingly ambitious and original visual effects created for non-Hollywood productions.

London-based Aardman Studios continued their relationship with DreamWorks, and newcomer Vanguard Animation struck a multi-picture distribution deal with Disney for several computer-animated features. Framestore and the Computer Film Company (CFC), also in London, merged in 2001 and created memorable TV commercials for Microsoft's Xbox and Levi's jeans (Figs. 9.8.3, 13.3.1, and 13.6.1). Hong Kong–based Menfond and Centro Pictures continued

1.5.1 *Synchronicity* by Hans Uhlig was one of the earliest independent shorts to make extensive use of motion capture. (© 2000 Bay Vista Productions.)

1.5.2 The Nutcracker and Barbie strike a pose in this scene from *Barbie in the Nutcracker*, a direct-to-video computer -animated movie that combines motion capture with keyframe techniques. (BARBIE and associated trademarks and trade dress are owned by, and used with permission from, Mattel, Inc. © 2003 Mattel, Inc. All rights reserved.)

their innovative visual effects and computer animation work (Figs. 1.6.2, 13.1.2, and 13.7.1). In spite of increased production during these years the industry also went through a series of ups-and-downs, with significant layoffs everywhere. Fox Studios sold VIFX to Rhythm & Hues Studios, and The Walt Disney Company closed The Secret Lab, its visual effects group. In New York City R/Greenberg Associates closed its computer animation and visual effects division, and London-based Mill Film, the company that won a 2000 AMPAS award for visual effects in *Gladiator*, closed in 2002.

The game industry continued its competition with movies as the premier form of entertainment, with multiplayer online games gaining acceptance. In 2001 ***Lara Croft Tomb Raider*** became the highest-grossing movie based on a videogame. Sony's ***EverQuest***, for example, launched in 1999 and reported hundreds of thousands of users by 2003 (Fig. 4.7.3). Other multiplayer online games launched in 2002 include Electronic Arts' ***Ultima*** and ***Majestic***, and Maxis' ***The Sims***. ***URU: Ages Beyond Myst***, the online version of the popular game, was launched in 2003 (Fig. 4.7.4). The game platforms started to offer additional online services, and the new graphics cards for PCs fostered the development of innovative games by a wide number of developers and publishers. A few of the PC and platform games that gained critical recognition and/or economic success during 2000 include, for the PC ***Deus Ex*** by ION Storm Austin, ***The Sims*** by Maxis, ***Diablo II*** by Blizzard, and ***No One Lives Forever*** by Monolith; for the Dreamcast: ***Jet Grind Radio*** by Smilebit, ***Shenmue*** by Sega AM2, ***Samba de Amigo*** by SEGA Sonic Team, ***Seaman*** by Vivarium, and ***Crazy Taxi*** by Hitmaker; ***Spyro: Year of the Dragon*** by Insomniac for the PS2, and ***Legend of Zelda: Majora's Mask*** by Nintendo for the N64). In 2001 some of the games that received critical acclaim or public recognition include, for the PlayStation 2: ***Grand Theft Auto III*** by DMA Design-Rockstar Games, ***Jak & Daxter: The Precursor Legacy*** (Fig. 7.4.6), ***Final Fantasy X*** by Square, ***Ico*** from Sony Computer Entertainment, ***Rez*** from United Game Artists, and Konami's ***Metal Gear Solid 2: Sons of Liberty***; for the Xbox, ***Halo: Combat Evolved*** by Bungie Studios, ***Cel Damage*** from Pseudo Interactive, and ***Oddworld: Munch's Oddysee*** (Fig. 7.6.3); for the PC: ***Max Payne*** from Remedy Entertainment, and Activision's ***Return to Castle Wolfenstein***; and ***Black & White*** from Lionhead Studios for Dreamcast (Fig. 1.5.5).

The Visual Effects Society Award Categories

- Visual Effects in an Effects-Driven Motion Picture
- Supporting Visual Effects in a Motion Picture
- Visual Effects in a Television Miniseries, Movie, or Special
- Visual Effects in a Television Series
- Visual Effects in a Commercial
- Visual Effects in a Music Video
- Character Animation in a Live Action Motion Picture
- Character Animation in a Live Action TV Program, Music Video, or Commercial
- Character Animation in an Animated Motion Picture
- Special Effects in a Motion Picture
- Matte Painting in a Motion Picture
- Matte Painting in a TV Program, Music Video, or Commercial
- Models and Miniatures in a Motion Picture
- Models and Miniatures in a TV Program, Video, or Commercial
- Visual Effects Photography in a Motion Picture
- Effects Art Direction in a Motion Picture
- Effects Art Direction in a TV Program, Video, or Commercial
- Compositing in a Motion Picture
- Compositing in a TV Program, Music Video, or Commercial
- Performance by an Actor in an Effects Film

1.5.3 The twenty new categories that the Visual Effects Society (VES) introduced in 2002 to recognize the visual effects work from a variety of creative fields and media (winners at www.ves.org).

Visual Milestones: Late 2000s

The late 2000s had an abundance of visual effects blockbusters. To name a few: ***Pirates of the Caribbean*** with its innovative use of partial motion capture to animate pirate Davy Jones (2003, 2006 and 2007, Figs. 2.7.15 and 12.1.10); ***Transformers*** (2007) and its popcorn action; comicbook-inspired ***Constantine*** (2005); ***Superman Returns***, and ***X-Men: The Last Stand*** (2006); ***Spider-Man 3*** (2007); and ***The Incredible Hulk*** (2008). ***Hellboy II: The Golden Army*** and ***Iron Man*** (Fig. 2.7.7), both

also from 2008 and my personal favorites, offered creative, subtle, effective, and polished animation. The former had superb modeling craftsmanship and imaginative staging of animation, the latter had understated and elegant effects that complemented the tone and intention of the superhero storyline without calling attention to themselves.

Highly stylized animation and effects were offered in *Sin City* (2005), *300* (2007), *Speed Racer*, and *Wanted* (2008). Spectacular effects and character animation were seen in sequels and remakes including *The Chronicles of Narnia: The Lion, the Witch and the Wardrobe, King Kong, Star Wars: Episode III–Revenge of the Sith*; *Mission: Impossible III* (2006), *TMNT* (2007, Fig. 8.6.6), *Indiana Jones and the Kingdom of the Crystal Skull*, and *The Mummy: Tomb of the Dragon Emperor* (2008). A few examples of feature movies with outstanding "invisible" supporting visual effects include the historical reconstructions in *Memoirs of a Geisha* (2005); the fantastic dreams and nightmares in *The Science of Sleep* and *Pan's Labyrinth*, the birth scene in *Children of Men*, and the water simulations in *Poseidon* (2006); the action scenes in *The Kite Runner* and *The Bourne Ultimatum*, and the comedic touches in *Blades of Glory*, (2007).

The late 2000s had memorable contributions from DreamWorks with *Kung Fu Panda* and *Madagascar: Escape 2 Africa* (2008, Figs. 10.4.13 and 12.1.8). The latter continued the unique approach to comedic squash and stretch animation initiated with *Madagascar* in 2005. Fox's Blue Sky continued with good-quality sequels to their *Ice Age* franchise, along with an interesting version of *Horton Hears a Who!* (2008). Sony Pictures Animation (SPA) delivered animated features produced in-house including *Open Season* (2006, Fig. 10.4.3) and *Surf's Up* (2007); *Open Season 2* (2009) was subcontracted to an outside production company. Sony Imageworks delivered a successful combination of motion capture and keyframe animation with *Monster House*, nominated in 2007 for an Academy Award and directed by first-timer Gil Kenan, with a fresh animation and cinematography style (Fig. 12.2.3). *Beowulf* (2007) also provided solid storytelling with ambitious computer animation based on performance capture.

Pixar continued their inimitable style and track-record with my personal favorite *Ratatouille* (2007), stylistically different *WALL·E* (2008), and *Cars* (2006). *Chicken Little* (2005) was Disney's much awaited debut with an all-computer-animated feature movie. This was followed-up with *Meet the Robinsons* (2007) and *Bolt* (2008), the latter released in stereo 3D day-and-date along with the theatrical release. *The Princess and the Frog* (2009) is Disney's first two-dimensional animated feature produced since 2004. Interesting approaches to low-budget animated features and a respectable box-office return for some is found in independent productions *Hoodwinked!* (2005), *Happily N'Ever After*, *The Barnyard*, and *The Wild* (2006).

Visual effects production continued to grow internationally and became more accomplished during this period. Take, for example, the French-produced *Empire of the Wolves, Arthur and the Invisibles*, and *Asterix in the Olympic Games*, the Chinese-produced *The Promise*,

1.5.4 Jimmy Neutron and friends having a snack. (© 2003 Viacom Inc. All rights reserved. Nickelodeon, *The Adventures of Jimmy Neutron Boy Genius* and all related titles, logos and characters are trademarks of Viacom International Inc.)

1.5.5 A character in the game *Black &White*. (© Lionhead Studios.)

1.5.6 Main character in *Dreammaker* (© 2007 Filmakademie Baden-Württemberg, and Leszek Plichta.)

1.5.7 Still frame from Indian feature *Ghatothkach*. (© Copyright 2008 Shemaroo Entertainment Pvt. Ltd. All rights reserved.)

1.6.1 (Opposite page) Color coding for each of the categories in the timeline charts on the following pages. In green are the live action feature movies with computer-generated visual effects. In **magenta** are animated feature movies, with a focus on three-dimensional computer animation. Winners of the Academy of Motion Picture Arts and Sciences (AMPAS) awards in the categories of Best Visual Effects and Best Animated Feature are marked with an asterisk, the runners-up in each category are listed above the thin black line. Independent productions and short computer animations are in **blue** boxes. Video, computer, and platform games are in **brown**. A selection of computer technology milestones and related industry and business events are grouped under **red**. A variety of related events and technologies and facts are indicated in **yellow**. Television animated series and events are presented in **deep blue**.

Japanese ***Shinobi*** (Fig. 2.7.6), Russian ***Day Watch***, and Korean ***The Host***. Highlights of international computer-animated features include Australian 2006 Oscar-winning ***Happy Feet***, French black-and-white movies ***Renaissance*** (2006) and ***Fear(s) of the Dark*** (2007, Fig. 10.3.7), and ***Dragon Hunters*** (2008, Fig. 11.1.0), Nordic ***The Ugly Duckling and Me!*** and ***Free Jimmy*** (2006, Fig. 6.1.4), Spanish ***Donkey Xote*** (2007, Fig. 7.2.8), Japanese ***Final Fantasy VII: Advent Children***, non-photorealistic ***Tekkonkinkreet*** (2006), and well-crafted ***Appleseed Saga Ex Machina*** (2007). We can also mention ***Fly Me to the Moon***, a Belgian production developed from start for 3D stereo projection, Indian ***Ghatothkach*** (2008, Fig. 1.5.7), and English-produced ***The Tale of Despereaux*** (2008). Highlights of independently produced animated shorts include SIGGRAPH winners ***9*** (2005) by Shane Acker scheduled to be released as a feature film in 2008, ***458nm*** by Jan Bitzer and others (Fig. 14.1.4) and Alex Weil's ***One Rat Short*** (2006); ***Ark*** by Grzegorz Jonkajtys and Marcin Kobylecki (Fig. 4.5.6) and ***Dreammaker*** by Leszek Plichta (2007, Fig. 1.5.6); and Academy Award nominees ***Oktapodi*** (2008) by Julien Bocabeille and others, and ***Even Pigeons Go to Heaven*** (2005) by BUF Compagnie (Figs. 7.3.4 and 10.3.5).

The remaining years of this decade promise to be as active as the early ones. As reported in the trade periodicals several animated and VFX feature films are in production as this book goes to press. Visual effects movies released or scheduled to be released throughout 2009 and 2010 include ***2012*** directed by Roland Emmerich, the futuristic epic ***Avatar*** directed by James Cameron, ***G.I. Joe: Rise of Cobra***, ***The Green Hornet*** directed by Stephen Chow, ***Harry Potter and the Half Blood Prince***, ***Iron Man II***, ***Kung Fu Hustle 2***, ***Prince of Persia***, ***Sin City 2***, ***Star Trek*** AKA ***Star Trek Zero***, ***Taken***, ***Terminator Salvation: The Future Begins***, ***Transformers 2: Revenge of the Fallen*** directed by Michael Bay, ***Watchmen***, ***Where the Wild Things Are*** directed by Spike Jonze, and ***X-Men Origins: Wolverine***. The working titles of a few animated projects scheduled for a 2009 or 2010 release include ***1906*** directed by Brad Bird, the much-anticipated ***Astro Boy*** being produced in Hong Kong by Imagi, ***Guardians of Ga'hoole***, ***Hoodwinked 2: Hood vs. Evil***, from Blue Sky/Fox ***Ice Age: Dawn of the Dinosaurs***, Spanish-produced ***Planet 51***, DreamWorks' ***Shrek Goes Fourth***, ***Toy Story 3***, DreamWorks' ***Monsters vs. Aliens***, French-produced ***A Monster in Paris***, and ***Up*** co-directed by Pete Docter and Bob Peterson at Pixar.

1.6 Timeline Charts

The following timeline charts offer a chronological overview of the development of three-dimensional computer animation and related events during the last four decades of the twentieth century. The charts are limited to selected events and landmarks, and additional historical details can be found in the body of this chapter or at the website *www.artof3d.com*. To make the information easier to read I have organized the charts in five categories, explained in Figure 1.6.1.

1890s–1950s Timeline of Animation and Visual Effects

1892...	1914...	1927...	1932...	1939...
Edison and W. Dickson develop the **35 mm** format using Eastman Kodak film stock, 1892 ***The Execution of Mary, Queen of Scots***, 1895 Lumiere Brothers' ***Train Arriving at Station***, 1895 George Méliès ***A Trip to the Moon***, 1902 Edwin Porter's ***The Great Train Robbery***, 1903 Winsor McCay's ***Little Nemo*** is first animated short, 1911	Winsor McCay's ***Gertie The Dinosaur***, first short with live action and animation, 1914 D. W. Griffith's ***The Birth of a Nation***, 1915 Max Fleischer invents **rotoscoping**, 1915 Pathé Baby introduces **9.5 mm** film, 1922 Kodak introduces **16 mm** reversal film, 1923 First Version of ***The Thief of Bagdad***, 1924 Willis O'Brien's animates dinosaurs in ***The Lost World***, 1925	Lang's ***Metropolis***, '26 ***The Jazz Singer*** first talkie film, 1927 Disney's ***Steamboat Willie*** is first animated cartoon with synchronized sound, 1928 ***The Skeleton Dance*** by Ub Iwerks, first Disney ***Silly Symphony***, 1929 Fleischer Brother's ***Betty Boop***, 1930 **TV set** patented, 1930 ***Frankenstein***, 1931 Disney's ***Flowers and Trees*** is first color animated short, 1932	Kodak's **8 mm** film and equipment, 1932 ***King Kong***, and ***The Invisible Man***, 1933 Fleischers' ***Popeye the Sailor*** debuts, 1933 ***The Bride of Frankenstein***, 1935 ***Things to Come***, 1936 ***Snow White and the Seven Dwarfs***, first animated feature, 1937 Academy of Motion Picture Arts and Sciences creates Special Effects category, 1939	VFX Movies 1939: ***The Rains Came **** ***Gone with the Wind*** ***Only Angels Have Wings*** ***The Private Lives of Elizabeth and Essex*** ***Topper Takes a Trip*** ***Union Pacific*** ***The Wizard of Oz*** Fleischer Brothers' ***Gulliver's Travels*** and ***Felix the Cat*** 1939 Disney's ***Pinocchio,*** and ***Fantasia,*** 1940 (* AMPAS Award Winners)

1940–41	1941–42	1942–43	1944–45	1945–47
VFX Movies 1940: ***The Thief of Bagdad **** 13 other finalists, incl.: ***Dr. Cyclops*** ***Invisible Man Returns*** ***Rebecca / Typhoon*** ***Swiss Family Robinson*** VFX Movies 1941: ***I Wanted Wings **** 7 other finalists, incl.: ***Flight Command*** ***The Invisible Woman*** ***The Sea Wolf*** Other Movies w/VFX: ***Citizen Kane***	Disney's ***Dumbo,*** 1941 **Superman** animated series debuts with a 9 minute episode, 1941 VFX Movies 1942: ***Reap the Wild Wind **** 9 other finalists, incl.: ***The Black Swan*** ***Flying Tigers*** ***One of our Aircraft is Missing*** ***Invisible Agent***	Disney's ***Bambi,*** 1942 Paul Terry creates **Mighty Mouse** as a **Superman** spoof, 1942 VFX Movies 1943: ***Crash Dive **** 5 other finalists, incl.: ***Air Force*** ***Bombardier*** ***The North Star*** Disney's ***Saludos Amigos***, 1943	VFX Movies 1944: ***Thirty Seconds Over Tokyo **** 6 other finalists, incl.: ***The Adventures of Mark Twain*** ***Secret Command*** VFX Movies 1945: ***Wonder Man **** ***Captain Eddie*** ***Spellbound*** ***They Were Expendable*** ***A Thousand and One Nights***	Disney's ***The Three Caballeros,*** 1945 VFX Movies 1946: ***Blithe Spirit **** ***A Stolen Life*** VFX Movies 1947: ***Green Dolphin Street **** ***Unconquered*** Disney's ***Make Mine Music***, 1946 Disney's ***Fun and Fancy Free,*** 1947

1948–49	1950–52	1953–55	1955–57	1958–59
VFX Movies 1948: ***Portrait of Jennie **** ***Deep Waters*** Disney's ***Melody Time,*** 1948 15-episode live action **Superman**, 1948 VFX Movies 1949: ***Mighty Joe Young **** ***Tulsa*** Disney's ***The Adventures of Ichabod and Mr. Toad***, 1949	VFX Movies 1950: ***Destination Moon **** ***Samson and Delilah*** VFX Movies 1951: ***When Worlds Collide **** (only nominee) Disney's ***Cinderella,*** 1950 Disney's ***Alice in Wonderland***, 1951 VFX Movies 1952: ***Plymouth Adventure **** (only nominee)	VFX Movies 1953: ***War of the Worlds **** (only nominee) Disney's ***Peter Pan,*** 1953 VFX Movies 1954: ***20,000 Leagues Under the Sea **** ***Hell and High Water*** ***Them!*** ***Godzilla*** AKA ***Gojira***	VFX Movies 1955: ***The Bridges At Toko-Ri **** ***The Dam Busters*** ***The Rains of Ranchipur*** Disney's ***Lady and the Tramp***, 1955 VFX Movies 1956: ***The Ten Commandments **** ***Forbidden Planet*** VFX Movies 1957: ***The Enemy Below **** ***The Spirit of St. Louis***	VFX Movies 1958: ***Tom Thumb **** ***Torpedo Run*** ***Vertigo*** VFX Movies 1959: ***Ben Hur **** ***Journey to the Center of the Earth*** Disney's ***Sleeping Beauty***, 1959 Ub Iwerks improves **optical film printer** to shoot successive exposures, 1959

1960s Timeline of Animation and Visual Effects

1960

Visual Effects Movies:
***The Time Machine* ***
The Last Voyage

Psycho
Spartacus

John McCarthy develops the **LISP** programming language for artificial intelligence (AI) applications.

Kodak introduces the **Ektachrome 7386** reversal print film stock.

Warner Bros.' (WB) ***The Bugs Bunny Show*** animated series debuts on ABC primetime.

Hanna-Barbera's ***The Flintstones*** is the first animated series to debut on primetime TV (ABC, 166 episodes).

1961

Visual Effects Movies:
***The Guns of Navarone* ***
The Absent Minded Professor

El Cid

Disney's ***One Hundred and One Dalmatians***

John Whitney creates ***Catalog*** on 16 mm film and a mechanical analog computer.

Fairchild Camera and Semiconductor Instrument manufactures the first integrated circuit on a chip.

Hanna-Barbera's (HB) ***Top Cat*** animated series debuts on ABC.

1962

Visual Effects Movies:
***The Longest Days* ***
Mutiny on the Bounty

Dr. No (first 007 film)

Sketchpad system for interactive computer graphics developed by Ivan Sutherland at MIT.

MIT students Slug Russell, Shag Graetz, and Alan Kotok create ***SpaceWar!*** the first interactive computer game on a DEC PDP-1.

Mr. Computer Image ABC, created by Lee Harrison III with the Scanimate System.

Hanna-Barbera's futuristic ***The Jetsons*** animated series debuts on ABC (24 episodes).

1963

Visual Effects Movies:
***Cleopatra* ***
The Birds

Jason and the Argonauts
From Russia with Love

The stop-motion skeletons animated by Ray Harryhausen in ***Jason and the Argonauts*** become effects classic.

Disney's ***The Sword in the Stone***

John Whitney's ***Lapis***

IBM introduces the 360 models, the first family of computers, a Fortran-based time-sharing system.

Polaroid introduces instant color film.

1964

Visual Effects Movies:
***Mary Poppins* ***
7 Faces of Dr. Lao

Goldfinger

Mary Poppins live action movie with 2D animated sequences.

Atboftb Carrier Landing, a 3D animation by William Fetter at Boeing in Seattle

Thomas Kurtz and John Kemeny develop the **BASIC** programming language.

Instant replay and slow motion debut on televised sports.

Hanna-Barbera's sci-fi action-adventure ***Jonny Quest*** debuts on ABC (26 episodes).

1965

Visual Effects Movies:
***Thunderball* ***
The Greatest Story Ever Told

Thunderball (fourth 007 film)

Stereo computer animations by Michael Noll and Bela Julesz at Bell Laboratories

Kodak introduces **Super 8 mm**, a new amateur film format.

Charlie Brown's Christmas debuts as the first animated special on television.

The Thunderbirds series breaks new ground in marionette TV animation.

1966

Visual Effects Movies:
***Fantastic Voyage* ***
Hawaii

Batman
The Battle of Algiers
Farenheit 451
One Million Years B.C.

Hummingbird by Charles Csuri, first examples of computer-generated representational animation

Final episode of the original ***The Flintstones*** series airs.

1967

Visual Effects Movies:
***Doctor Dolittle* ***
Tobruk

You Only Live Twice

Disney's ***The Jungle Book***

Cockpit Simulation by Boeing's W. Fetter with 3D CG human.

Rauschenberg and Klüver's **Experiments in Art and Technology** in New York City

Hanna-Barbera's ***The Fantastic Four*** and ***Shazzan*** debut on ABC.

Marvel Comics' ***Spiderman*** animated series debuts on ABC.

52 animated episodes of anime ***Speed Racer*** dubbed to English.

1968

Visual Effects Movies:
***2001: A Space Odyssey* ***
Ice Station Zebra

Planet of the Apes

New visual style in ***The Yellow Submarine*** by George Dunning

Permutations by John Whitney

Evans & Sutherland opens.

The Motion Picture Producers of America's (MPPA) new movie **rating system**

Kodak's **Eastman 5249** color reversal intermediate film, for one-step duplication of originals

WB releases ***Wacky Races*** animated series.

1969

Visual Effects Movies:
***Marooned* ***
Krakatoa, East of Java

On Her Majesty's Secret Service

A Boy Named Charlie Brown, by B. Melendez

Pas De Deux, by Norman McLaren

Sony's 3/4 in. **U-matic** video cassette

Kenneth Thompson and Dennis Ritchie develop **UNIX** at AT&T Bell Laboratories.

Warnock's **area subdivision** hidden surface removal algorithm

HB's ***Scooby-Doo Where are You!*** series

The ***Pink Panther Show*** animated series debuts on NBC.

1970s Timeline of Computer Animation and Visual Effects

Categories

- VFX Movies
- Animated Features
- Independent Shorts
- Computer Games
- Technology / Events
- Related Tech./Events
- Television

1970

Visual Effects Movies:
***Tora! Tora! Tora!* ***
Patton

Airport
M*A*S*H

Dr. Seuss' **Horton Hears a Who!**, half-hour animated TV special directed by Chuck Jones.

Minicomputers are this decade's tool of choice for 3D computer animation, including Digital Equipmrent Corporations's PDP and VAX models.

Watkins' **scan line** hidden surface removal

IMAX projection premieres at the Osaka Expo 70 in Japan.

1971

Visual Effects Movies:
***Bedknobs and Broomsticks* ***
When Dinosaurs Ruled the Earth

Silent Running

Fritz the Cat by Ralph Bakshi

Animated Faces, by Fred I. Parke at University of Utah

Intel releases its 4-bit **4004** microprocessor.

IBM invents the 8 in. floppy diskette.

Robert Abel & Associates opens.

Dolby techniques reduce noise in recorded sound in Kubrick's ***A Clockwork Orange.***

1972

Visual Effects Movies:
***The Poseidon Adventure* *** (only nominee)

Heavy Traffic by Ralph Bakshi

Atari opens and releases **Pong** arcade game.

Intel releases 8-bit **8008** microprocessor.

Newell's **depth sort** hidden surface removal

MAGI animates computer-rendered polygonal objects.

Phillips and MCA demonstrate **videodisc** recorder and player system.

Polaroid's **SX-70 camera** brings one-step instant photography.

1973

Visual Effects Movies:
The Exorcist
Westworld
(No AMPAS Visual Effects Award given this year).

The Savage Planet by René Laloux

Warner Bros. and Hanna-Barbera's ***Charlotte's Web***

Hanna-Barbera releases ***The Adams Family*** animated TV series.

Cable TV goes into the mainstream during this decade.

*Toei Animation releases **Mazinger Z** TV animated series.*

1974

Visual Effects Movies:
***Earthquake* *** (only nominee)

Chinatown
The Towering Inferno
Young Frankenstein

Hunger by Peter Foldes, National Film Board of Canada

Intel and Zilog release the **Intel 8080** microprocessor.

The first **SIGGRAPH** conference is held in Boulder, Colorado, with 600 attendees.

Technicolor stops U.S. production of **dye-transfer** film prints.

1975

Visual Effects Movies:
***The Hindenburg* *** (only nominee)

Jaws
Rollerball

George Lucas starts **Industrial Light & Magic** in California.

Sony introduces the 1/2 in. **Betamax** videotape format.

The University of Utah becomes a center of innovative research in computer graphics.

1976

Visual Effects Movies:
***King Kong* ***
and ***Logan's Run* *** (dual award)

Shugart Associates develops the 5.25 in. **floppy diskette**.

The 64-bit **Cray I** supercomputer solves 166 million floating point operations per second.

Steve Wozniak designs the **Apple I.**

The **Steadicam**, a camera-stabilizing system, used for the first time in the film ***Rocky***.

Canon's **AE-1** is first 35 mm film still camera with a built-in microprocessor.

1977

Visual Effects Movies:
***Star Wars* ***
Close Encounters of the Third Kind

Airport '77
The Spy Who Loved Me

Voyager animated simulations of space exploration by Jim Blinn at Jet Propulsion Lab.

Atari introduces its **VCS 2600** videogame home system.

The **Apple II** microcomputer is released.

Radio Shack releases its **TRS-80** computer.

Matsushita releases the 1/2 in. **VHS** (Video Home System) videotape format.

1978

Visual Effects Movies:
***Superman* *** (only nominee)

Capricorn One
Halloween
Invasion of the Body Snatchers

The Lord of the Rings by Ralph Bakshi

Midway imports Taito's **Space Invaders** arcade game into U.S.

Digital Equipment releases the **VAX 11/780** minicomputer.

Artist **Leroy Neiman** uses a CG system to create 2D images in real-time during the CBS broadcast of the Super Bowl.

1979

Visual Effects Movies:
***Alien* ***
1941
The Black Hole
Moonraker
Star Trek–The Motion Picture

Apocalypse Now

Galaxy Express 999: The Signature Edition, anime by Rintaro

Hayao Miyazaki's ***The Castle of Cagliostro***

Atari releases **Asteroids** arcade game.

Motorola releases 32-bit **68000** processor.

Voyager 2 visualizations by James Blinn at the Jet Propulsion Lab

Digital Effects opens in New York, **Activision** in California, Computer group starts at **ILM**.

Early 1980s Timeline of Computer Animation and Visual Effects

1980

Visual Effects Movies:
***The Empire Strikes Back* ***
(only nominee)

Airplane!
Battle Beyond the Stars
Mad Max
Saturn 3
Xanadu

The Empire Strikes Back takes "effects movies" to a new level of complexity and accomplishment.

Paul Grimault's ***Le roi et l'oiseaux*** (The King and Mr. Bird).

Vol Libre, animation of fractal landscapes by Loren Carpenter

Atari releases ***Space Invaders*** for its VCS 2600 system.

Namco releases ***Pac Man***, most popular arcade game ever.

Seagate Technology releases **hard disks** for microcomputers.

Pacific Data Images (PDI) opens in Northern California.

Pioneer markets their **videodisc players**.

Microcomputers become popular during this decade.

1981

Visual Effects Movies:
***Raiders of the Lost Ark* ***
Dragonslayer

An American Werewolf in London
Clash of the Titans
Escape from New York
Looker
The Road Warrior

Looker, becomes first film featuring the first virtual actor, *Cindy*, made from simulated body scans of actress Susan Dey.

American Pop by Ralph Bakshi

The Secret of Nimh, by Bluth Productions

IBM releases its 8088-based **PC computer** with the MS-DOS operating system.

Sony introduces 3.5 in. diskettes.

Philips develops the optical **CD-ROM**.

Adam Osborne develops the 24 lb. **Osborne I**, the first portable computer.

Wavefront opens in Santa Barbara, **Digital Productions** in L.A., **R/Greenberg Assoc.** (RGA) in New York, and **Cranston/Csuri Productions** in Columbus, Ohio.

Quantel introduces its **Paintbox** digital paint system.

MTV goes live on cable television.

Hanna-Barbera's ***The Smurfs*** series debuts on NBC.

1982

Visual Effects Movies:
***E.T. the Extra-Terrestrial* ***
Blade Runner
Poltergeist

Firefox
Star Trek II–The Wrath of Kahn
TRON

TRON is first live action film with over 20 minutes of 3D computer animation.

ILM's **Genesis Effect** created for ***Star Trek II***, is the first all-computer-animated visual effects movie shot.

Jim Henson creates ***The Dark Crystal*** with puppetry, stop motion, and animatronics.

Le maitres du temps by René Laloux

Carla's Island by Nelson Max at the Lawrence Livermore National Lab

SubLogic develops the **Microsoft Flight Simulator** computer game for the Apple II, later for the PC.

CT5 Flight Simulator by Evans & Sutherland

Non-Edge Cloud and Smoke Simulations by Goffrey Gardner at Grumman Systems

Autodesk opens in California, releases **AutoCAD** for the PC.

Silicon Graphics (SGI), **Adobe**, and Electronic Arts (EA) open in California; **Omnibus Computer Graphics** and **Alias Research** in Toronto; **Toyo Links** opens in Tokyo.

Canon demonstrates first **electronic still camera**.

1983

Visual Effects Movies:
***Return of the Jedi* ***
(only nominee)

Blue Thunder
Brainstorm
Hercules
Jaws 3-D
Octopussy
Superman III
Twilight Zone: The Movie
WarGames

Fire and Ice by Ralph Bakshi.

Growth: Mysterious Galaxy, the first in a series of semi-abstract animations by artist-programmer Yoichiro Kawaguchi.

The inexpensive **Commodore 64** computer outperforms videogame consoles.

Arcade game **Dragon's Lair**, animated by Don Bluth, first to use laserdisc technology.

Compaq introduces the first **PC-clone** computer.

SGI introduces the **IRIS 1000** computer based on a Motorola 68000 with a dozen **Geometry Engines**.

Tippett Studios opens in Berkley, **Polygon Pictures** in Tokyo.

During the mid-1980s commercial production is led by Pacific Data Images (PDI), Digital Productions, Cranston-Csuri, Sogitec, Toyo Links and Omnibus.

MIDI (Musical Instrument Digital Interface) introduced by electronic instrument manufacturers.

1984

Visual Effects Movies:
***Indiana Jones and the Temple of Doom* ***
Ghostbusters
2010

Nausicaä of the Valley of the Wind by Hayao Miyazaki.

Trailer for ***The Works***, an unfinished New York Institute of Technology movie.

Bio-Sensor created at Osaka University and Toyo Links, is an early example of modeling with blobby surfaces and figure locomotion.

Still Life Etude, an early simulation of light, fog, rain, and skies created at Hiroshima University.

Apple Computer releases the **Macintosh**.

IBM releases its 80286-based **PC-AT**.

Pixar opens its doors.

Film Roman opens in L.A., The **Computer Film Company** (CFC) in London.

MPPA rating system is expanded to include a **PG-13** category.

Herbie Hancock's music video ***Rockit*** receives Best Special Effects award at MTV's first Video Music Awards.

Hanna-Barbera's ***The New Scooby-Doo Mysteries*** TV series debuts on ABC.

Late 1980s Timeline of Computer Animation and Visual Effects

Categories
- VFX Movies
- Animated Features
- Independent Shorts
- Computer Games
- Technology / Events
- Related Tech./Events
- Television

1985

Visual Effects Movies:
***Cocoon* ***
Return to Oz
Young Sherlock Holmes

Back to the Future
The Last Starfighter

The Last Starfighter created with a Cray supercomputer at Digital Productions, first live action feature film with realistic computer animation.

The Black Cauldron, first Disney animated feature film to use 3D CG technology

Brilliance commercial by Abel and Associates features sexy female robot with convincing realistic motion.

Growth III by Yoichiro Kawaguchi

Tony de Peltrie by Pierre Lachapelle and team.

U.S. release of 8-bit ***Nintendo Entertainment System (NES)*** with Super Mario Brothers.

SEGA releases 8-bit ***Master*** game system.

Bell Labs' Bjarne Stoustrup develops **C++** prog. language.

Kleiser-Walczak opens. **Digital Effects** closes. **The Moving Picture Co.** (MPC) starts 3D production in London.

First wave of user-friendly 3D computer animation software.

The National Science Foundation creates **NSFNET**, a 56 Kbps network (future Internet), for research and academic work.

1986

Visual Effects Movies:
***Aliens* ***
Little Shop of Horrors

Flight of the Navigator

In Disney's ***The Great Mouse Detective*** the moving gears in the chase sequence are created with 3D computer animation.

Laputa: Castle in the Sky by H. Miyazaki

An American Tail by Don Bluth

Jim Henson's ***Labyrinth***

John Lasseter's ***Luxo Jr.*** nominated in AMPAS Animated Short Films category.

The still image ***Road to Point Reyes*** redefines realism by compositing graftals and fractals to portray a landscape.

Visitor on a Foggy Night by the CG Research Group at Hiroshima University

Intel releases the 32-bit, 4-million-operations-per-second **Intel 80386** microprocessor.

SGI introduces the **IRIS 3000** series, with a MIPS single processor, and a 10 MHz **Geometry Engine**.

Softimage opens in Montreal, **Mac Guff Ligne** in Paris, **Ubisoft** in Montreuil-sous-Bois, **VIFX** in L.A., **Mental Images** in Berlin, and **Framestore** in London. **Omnibus** declares bankruptcy.

Quantel's **Harry** provides component digital video processing.

Toei Animation releases ***Dragon Ball*** series.

1987

Visual Effects Movies:
***Innerspace* ***
Predator

Akira by Katsuhiro Otomo popularizes anime, feature-length sci-fi Japanese animation, with international audiences.

Stanley and Stella: Breaking the Ice by Symbolics Graphics and Whitney Demo Productions, early flock animation

Red's Dream by Pixar

Baloon Guy by Chris Wedge at Ohio State University

Rendezvous in Montreal by Nadia Magnenant Thalmann and team.

Nintendo releases ***Legend of Zelda*** NES cartridge in the U.S.

Rhythm & Hues opens in Los Angeles, **Blue Sky Studios** opens in New York, **Side Effects Software** in Toronto.

Research in the simulation of natural-looking hair and fur, rigid body dynamics, and modeling fabric with visible threads starts in the late 1980s.

First **Game Developers Conference** organized by Chris Crawford in San Jose, California, with 27 attendees.

Bill Atkinson at Apple Computer develops **Hypercard** as an interactive software development tool.

Hanna-Barbera releases 13 new episodes of ***Jonny Quest***.

1988

Visual Effects Movies:
***Who Framed Roger Rabbit* ***
Die Hard
Willow

Who Framed Roger Rabbit breaks new ground by combining live actors with animated characters.

Many cars in Disney's ***Oliver & Company*** are 3D CGI models.

My Neighbor Totoro by Hayao Miyazaki

Stop motion, puppetry, and live action in Jan Svankmajer's ***Alice***

The Land Before Time by Don Bluth, the first of many in a direct-to-video franchise

Graveyard of the Fireflies by anime director Isao Takahata

Pixar's ***Tin Toy*** by John Lasseter and William Reeves wins AMPAS award.

Technological Threat by William Kroyer and Brian Jennings

Locomotion, a Pacific Data Images short, early example of 3D squash-and-stretch.

Nintendo releases portable ***Game Boy*** at 140 × 120 pixels, 2.14 MHz, with ***Tetris*** game.

RenderMan shading language released and awarded U.S. patent.

Namco purchases the Japan Computer Graphics Lab.

The Sky, simulations of light and skies at Hiroshima University.

Waldo C. Graphic animated in real time for the ***Jim Henson Hour***.

1989

Visual Effects Movies:
***The Abyss* ***
The Adventures of Baron Munchausen
Back to the Future Part II

Indiana Jones and the Last Crusade
Field of Dreams
Ghostbusters II

The Abyss includes first convincing 3D character animation.

The Little Mermaid is Disney's last film to use traditional ink and paint, CG closing shot.

Kiki's Delivery Service by Hayao Miyazaki opens in Japan.

All Dogs Go to Heaven by Don Bluth

Knickknack by Pixar, ***Don't Touch Me*** by Kleiser-Walczak, early motion capture character animation

Preview of NYIT's ***The Works***, ***The Little Death*** by Matt Elson at Symbolics, ***Eurythmy*** by Susan Amkraut and Michael Girard

SEGA releases 16-bit ***Genesis*** game system.

Maxis releases ***SimCity***.

Intel releases its 32-bit **80486** microprocessor.

CA Scanline opens. Sogitec and TDI merge into **Ex Machina.**

Letraset releases **Color Studio** image retouching software for Macs.

Matt Groening's ***The Simpsons*** TV debut.

Early 1990s Timeline of Computer Animation and Visual Effects

1990

Visual Effects Movies:
Total Recall *
(only nominee)

Back to the Future III
Die Hard 2: Die Harder
Dick Tracy
Ghost
The Hunt for Red October

Disney's ***The Rescuers Down Under***, first Disney animated feature film done entirely with the first version of the CAPS System

Hanna-Barbera's ***Jetsons: The Movie*** includes computer animated vehicles and environments.

Karl Sims' ***Panspermia***, early particle systems computer animation.

Origin Systems releases ***Wing Commander***, a cinematic space adventure PC game.

Nintendo releases ***Super Mario 3***, the all-time best-selling video-game cartridge.

Adobe releases **Photoshop** for Apple Macintosh computers.

NewTek's **Lightwave** software bundled with ***Toaster*** hardware.

Santa Barbara Studios opens in Los Angeles, **The Mill** in London.

Tim Berners-Lee at CERN develops the HyperText Markup Language (**HTML**).

MPAA renames their X rating to **NC-17** (No one 17 and Under).

Warner Bros. releases ***Steven Speilberg's Tiny Toon Adventures*** animated series.

1991

Visual Effects Movies:
Terminator II: Judgment Day *
Backdraft
Hook

Star Trek VI

Terminator II is the first mainstream blockbuster movie with multiple morphing effects and simulated natural human motion.

The animated camera in Disney's ***Beauty and the Beast*** travels in 3D space; first animated film nominated for AMPAS Best Picture.

Studio Ghibli's ***Only Yesterday*** AKA ***Omohide Poro Poro***

Mutations by William Latham and IBM UK

Don Quichotte by Video System uses keyframe character animation techniques.

Leaf Magic by Alan Norton uses motion dynamics animation.

U.S. release of 16-bit ***Super Nintendo Entertainment System (SNES)*** with real-time scaling, transparency.

Spectrum Holobyte's ***Falcon***, multiplayer jet combat simulation game. Capcom's ***Street Fighter II*** arcade game.

Motorola's 32-bit **68040** microprocessor.

Apple Computer releases **QuickTime.**

LINUX v. 0.01 Open Source OS released to Net by Linus Torvalds.

Animal Logic opens in Sydney, Australia, **Cinesite** in Hollywood, **Discreet Logic** in Montreal.

Disney/Pixar $26 million deal to produce 3 computer-animated features.

1992

Visual Effects Movies:
Death Becomes Her *
Aliens 3
Batman Returns

Bram Stoker's Dracula
The Lawnmower Man

Early 1990s defined by succesful revival of live action feature movies with visual effects.

Aladdin is Disney's first use of fully computer animated character and 3D organic surfaces.

Porco Rosso by Hayao Miyazaki

Kroyer Films' hand-drawn ***Fern Gully... The Last Rainforest*** uses edge-detection filters to draw outlines around 3D objects.

Cool World by Ralph Bakshi

Liquid Selves, particle systems animation by Karl Sims

The Seven Wonders of the World by Electric Images pushes the boundaries of architectural visualization.

Foundation Imaging and **Sony Imageworks** open.

SimmGraphics' facial motion capture system **Facetracker** used to animate ***Super Mario.***

NSFNET upgrades to 45 Mbps **T-3** lines.

Warner Bros. releases ***Batman: The Animated Series*** and the ***Steven Spielberg Presents Animaniacs*** series.

(1991)
Kodak's **Professional Digital Camera System**

Peter Greenaway's ***Prospero's Books*** integrates windows with traditional film editing.

1993

Visual Effects Movies:
Jurassic Park *
Cliffhanger
The Nightmare Before Christmas

The Fugitive

Jurassic Park sets new standards for realism, inverse kinematics and digital compositing.

Tim Burton's ***The Nightmare Before Christmas*** takes stop motion to new heights and becomes a classic.

Computer animation for ***Babylon 5*** TV series is produced with off-the-shelf microcomputer systems.

First ***Polar Bears*** commercial for Coca-Cola by Rhythm & Hues.

Myst pushes the limits of a CD-ROM interactive visual experience.

George Romero and id Software release ***Doom***. First-person shooting games are forever changed.

Pixar receives an AMPAS Technical Award for RenderMan.

Adobe releases Windows **Photoshop**.

SGI's **Onyx** with 2–24 MIPS R-4400 procs and **Reality Engine2**.

Digital Domain and **Nvidia** open in Calif., **Kaydara** in Montreal, **Weta Digital** in New Zealand, **Mainframe** in Vancouver, Canada.

Cartoon Network goes live with ***The Moxy Show***, an early real-time virtual TV host.

WB releases the ***Two Stupid Dogs*** TV series.

WB releases the ***Batman: Mask of the Phantasm*** in theatres instead of DTV.

1994

Visual Effects Movies:
Forrest Gump *
The Mask
True Lies

The Flintstones
Speed

The wildebeest stampede in Disney's ***The Lion King*** is a tour de force in the integration of traditional and 3D computer animation.

Jan Svankmajer's ***Faust***

Don Bluth's ***Thumbelina***

Listerine Arrows TV commercial by Pixar

Sony's **PlayStation** and SEGA's **Saturn** 32-bit platforms are introduced in Japan (U.S. in '95), reinvigorate electronic game industry.

The Entertainment Software Rating Board issues **rating categories** for video and computer games.

CORE Digital Pictures opens in Toronto, **Cinesite** in London.

Immersion, an early experiment in image-based rendering

Supermicrocomputers, or workstations, based on 32-bit or 64-bit CISC and RISC processors gain popularity early in the decade.

Apple releases ***PowerPC*** computers.

NewTek starts selling **Lightwave** as stand-alone software.

Softimage ships **Mental Ray** as optional renderer.

Microsoft buys Softimage.

DVD format debuts.

Late 1990s Timeline of Computer Animation and Visual Effects

Categories
- VFX Movies
- Animated Features
- Independent Shorts
- Computer Games
- Technology / Events
- Related Tech./Events
- Television

1995

Visual Effects Movies:
Babe* *
Apollo 13

Batman Forever
Casper / ***Congo***
The City of Lost Children AKA ***La cité des enfants perdus***
Crimson Tide
Die Hard: With a Vengeance
Goldeneye
Judge Dredd
Jumanji / ***Stargate***
Species
Twelve Monkeys
Waterworld

Goldeneye is first 007 film with 3D computer-animated effects.

Toy Story is first fully 3D computer-animated feature movie.

The canoe and Mother Willow sequences in Disney's ***Pocahontas*** created with 3D CG.

Mamoru Oshii's ***Ghost in the Shell*** (U.S. in '98)

Squash and stretch in R/GA's ***Dance Fever*** cartoon car commercial

Chris Landreth's ***the end***

id Software releases online game ***Quake.***

Nintendo 64 64-bit platform introduced in Japan (U.S. in '96).

Blur and **Pixel Liberation Front** open in Venice, California, **Sparx*** opens in Paris.

Alias/Wavefront bought by Silicon Graphics.

The **Internet** becomes a self-supporting commercial operation.

DV videotape format introduced by consortium of 55 companies.

MPEG-2 format and spec published

Warner Bros. releases the ***Pinky and the Brain*** TV series.

1996

Visual Effects Movies:
Independence Day* *
Dragonheart
Twister

Mission Impossible
The Rock
Star Trek: First Contact

James and the Giant Peach combines stop motion and computer animation techniques.

Disney's ***Hunchback of Notre Dame*** has 3D confetti, crowds, and architectural details.

Warner Bros.'s ***Space Jam*** features the Loony Tunes characters.

The Fight by Acclaim Entertainment proves viability of mocap for character animation.

Joe's Apartment Roach Rally by Blue Sky Productions.

Nintendo's ***Super Mario 64*** (N64), Capcom's ***Resident Evil*** (PS), Namco's arcade ***Soul Edge***. Core Design's ***Tomb Raider***.

Dreamworks buys an interest in **Pacific Data Images**. Disney buys **Dream Quest Images.** Fox purchases **VIFX**.

Microsoft releases **Windows 95.**

Visual Effects Society (VES) is founded.

Hanna-Barbera's ***Dexter's Laboratory*** and ***The Real Adventures of Jonny Quest*** series debut on Cartoon Network.

Warner Bros. releases ***Superman: The Animated Series.***

(1997)

Pixologic opens in Los Angeles, Calif., and releases **ZBrush**.

1997

Visual Effects Movies:
Titanic* *
The Lost World: Jurassic Park
Starship Troopers

Air Force One
Alien: Resurrection
Batman & Robin
Con Air / ***Contact***
Dante's Peak
The Fifth Element
Flubber
Mars Attacks!
Men in Black
Spawn / ***Volcano***

Princess Mononoke by Hayao Miyazaki opens in Japan (U.S in '99).

Hydra sequence and morphed clouds in Disney's ***Hercules***.

Don Bluth's ***Anastasia***

I Married a Strange Person by Bill Plympton

Megasónicos is first European 3D computer-animated feature.

Pixar's ***Geri's Game*** by Jan Pinkava wins AMPAS short award.

Virtual Andre commercial by Digital Domain uses mocap.

Namco's ***Soul Blade*** weapon-fighting game.

Activision's ***Mech Warrior*** (PS). Rare's ***Golden Eye 007*** (N64).

Capcom's ***Street Fighter III*** and Namco's ***Tekken III*** to arcades.

Nvidia launches a 128-bit Direct3D processor.

News Corp. Fox buys **Blue Sky Studios**.

Sony introduces the **HDCAM** format.

South Park and ***King of the Hill*** premiere on MTV and Fox. HB's ***Cow and Chicken*** and ***Johnny Bravo*** debut on Cartoon Network.

Mainframe's ***ReBoot*** is early all-CG TV series.

1998

Visual Effects Movies:
What Dreams May Come* *
Mighty Joe Young
Armageddon

Deep Impact
Deep Rising
Godzilla
Lost in Space
Mouse Hunt
Pleasantville
Small Soldiers
Sphere / ***The X Files***

Disney/Pixar's ***A Bug's Life*** and DreamWorks/PDI's ***ANTZ*** present all-CG insect worlds.

CG Hun crowd simulation and CG props in Disney's ***Mulan***

Stylized characters in DW's ***Prince of Egypt***

Nickelodeon's low-budget ***Rugrats Movie*** is box-office success.

WB releases ***Quest for Camelot***.

S. Kon's ***Perfect Blue***

Cowboy Bebop AKA ***Kaubôi bibappu***

Kirikou et la Sorcière, by Michel Ocelot

Chris Wedge's ***Bunny*** wins AMPAS Award.

Bingo by Chris Landreth explores the absurd and neo-Dada theater.

Namco's ***Tekken 3*** (PS), Nintendo's ***Legend of Zelda: Ocarina of Time*** (N64), ***Quake 2*** for PC

Sega releases the **Dreamcast** platform.

Double Negative opens in London, **Realviz** in France. Avid buys Softimage.

Alias|Wavefront's **Maya**

Rolie Polie Olie all-CG 3D series premieres on The Disney Channel.

Hanna-Barbera's ***The Powerpuff Girls*** debut on Cartoon Network.

Kodak's **SFX 200T** film

1999

Visual Effects Movies:
The Matrix* *
Star Wars Episode I– The Phantom Menace
Stuart Little

End of Days
Fight Club
The Mummy
Sleepy Hollow
Wild Wild West

Toy Story 2 takes Buzz and Woody to new levels of comedic and technical achievement.

Disney's ***Fantasia 2000*** in IMAX, with 3D CG.

Fresh animation style and NPR rendering in Brad Bird's ***Iron Giant.***

2D brushstokes recreated on 3D geometry in Disney's ***Tarzan***.

South Park: Bigger, Longer & Uncut, uses 3D billboard technique.

Le Château des Singes, by Jean-F. Laguionie.

Daniel Robichaud's ***Tightrope***, playful jester confronts suit.

Bjork's ***All is Full of Love*** music video.

NPR rendering and surreal comedy of spatial errors in PDI's ***Fishing*** and ***Spatial Frames***

Fiat Lux by Paul Debevec, a landmark in image-based rendering

Piotr Karwas' ***The Mask*** receives first SIGGRAPH Jury Honors award.

SGI's **Pentium** workstations. Autodesk buys **Discreet Logic**.

Framestore's ***Walking with Dinosaurs***.

Fox's ***Futurama*** debut. WB's ***Batman Beyond***, Toei Animation's ***Digimon Adventure***

Star Wars: Episode I, Tarzan, and Miramax's ***An Ideal Husband***, early digital cinema.

Early 2000s Timeline of Computer Animation and Visual Effects

2000

Visual Effects Movies:
***Gladiator* ***
Hollow Man
The Perfect Storm

102 Dalmatians
The Adventures of Rocky & Bullwinkle
Cast Away
The Cell* / *Dinosaur
Dr. Seuss' How the Grinch Stole Christmas
Mission: Impossible 2
Mission to Mars
O Brother Where Art Thou?
Pitch Black
Red Planet* / *X-Men

Disney's ***Dinosaur*** combines live action backgrounds with realistic 3D computer animated characters.

Aardman Sudio/ DreamWorks stop-motion ***Chicken Run***

DreamWorks' ***The Road to El Dorado***

Don Bluth's ***Titan A.E.***

Disney's ***Emperor's New Groove*** and ***The Tigger Movie***

Nickelodeon's ***Rugrats in Paris***

Nintendo's ***Pokemon The Movie 2000*** and Toei Animation's ***Digimon: The Movie***

Pixar's ***For the Birds*** wins AMPAS award.

Onimusha wins Best Short at SIGGRAPH.

Alien Song is widely viewed on the Web.

Debut of **PlayStation 2**, **Xbox**, and **Gamecube**.

Maxis releases ***SimCity***, wins first GDC Best Game Award.

Intel's **Pentium 4** debuts at 1.5 GHz.

Nvidia's **GeForce2**, per-pixel shading GPU

SoftimageIXSI released.

Links DigiWorks opens in Tokyo.

2001

Visual Effects Movies:
***The Lord of the Rings: The Fellowship of the Ring* ***
Artificial Intelligence
Pearl Harbor

Aliens vs. Predator 2
Black Hawk Down
Cats & Dogs
Donnie Darko
Enemy at the Gates
Evolution
The Fast and the Furious
Harry Potter and the Sorcerer's Stone
Jurassic Park III
Lara Croft Tomb Raider
Monkeybone
Moulin Rouge!
The Mummy Returns
Ocean's Eleven
Planet of the Apes
Shaolin Soccer* AKA *Siu lam juk kau
Spy Kids
Swordfish

The AMPAS creates new category for Best Animated Feature.

Shrek combines irreverent comedy with cutting-edge rendering, wins AMPAS Animated Feature first award.

Pixar's ***Monsters, Inc.*** story turns the tables on who scares who.

Nickelodeon's ***Jimmy Neutron Boy Genius*** delivers using off-the-shelf software and below-average budget.

Square's ***Final Fantasy: The Spirits Within*** displays dazzling CG technique but fails to capture the mainstream box office.

Mainframe motion-captures American Ballet Theater dancers to animate Mattel's all-CG DTV ***Barbie in the Nutckracker***.

Metropolis, anime directed by Rintaro

Mamoru Oshii's ***Avalon***

2001 (cont.)

Animated Movies:
***Shrek* ***
Jimmy Neutron Boy Genius
Monsters, Inc.

Atlantis: The Lost Empire
Avalon
Final Fantasy: The Spirits Within
Marco Polo: Return to Xanadu
Mutant Aliens
Osmosis Jones
The Prince of Light
Recess: School's Out
The Trumpet of the Swan
Waking Life

Pixar short ***Mike's New Car*** by Pete Docter and Roger Gould.

Van Phan's ***Values*** Best Short at SIGGRAPH

Sega discontinues ***Dreamcast*** platform.

Game Developers Choice Awards (GDC) created by the International Game Developers Association (IGDA).

Rockstar's ***Grand Theft Auto III*** wins GDC Best Game Award.

Lionhead Studios' ***Black & White***; Bungie Studios' ***Halo: Combat Evolved***; Sony's ***Ico***, Remedy's Max Payne.

Framestore and **CFC** merge in London. **La Maison** opens in Paris.

GeForce3, Nvidia's programmable GPU.

Intel releases 64-bit ***Itanium*** processor.

Cartoon Network's ***Samurai Jack*** series. WB's ***Justice League***.

Foundation Imaging completes a US, $20 million, 26-episode CG series ***Dan Dare: Pilot of the Future*** based on the British 1950s comic book hero.

2002

Visual Effects Movies:
***The Lord of the Rings: The Two Towers* ***
Spiderman
Star Wars: Episode II - Attack of the Clones

28 Days Later...
Aliens vs. Predator 2: Primal Hunt
Astérix & Obélix: Mission Cléopâtre
Blade 2
Clockstoppers
Die Another Day
Eight Legged Freaks
Harry Potter and the Chamber of Secrets
Hero* AKA *Ying xiong
Men in Black 2
Minority Report
Panic Room
Reign of Fire
Resident Evil
Returner* AKA *Ritana
The Ring
Scooby-Doo
The Scorpion King
Signs* / *Solaris
Spy Kids 2: Island of Lost Dreams
Star Trek: Nemesis
Stuart Little 2
The Time Machine
xXx

The Lord of the Rings 2 combines keyframe techniques and performance capture to animate the *Gollum* character, superb crowd simulation software.

Star Wars: Episode II is shot on HD video with blue screen and virtual characters.

Robert Rodriguez's ***Spy Kids 2*** uses HD and desktop production.

Hayao Miyazaki's ***Spirited Away*** wins AMPAS award with fantastic story.

Fox's ***Ice Age*** by Chris Wedge mixes physical comedy with refined ray tracing rendering.

Mattel's direct-to-video ***Barbie as Rapunzel***

2002 (cont.)

Animated Movies:
***Spirited Away* ***
Ice Age
Lilo & Stitch
Spirit: Stallion of the Cimarron
Treasure Planet

Hey Arnold!
The Living Forest AKA ***El bosque animado***
Mutant Aliens
The Powerpuff Girls First Feature
Return to Never Land
The Wild Thornberrys

Disney's ***Lilo & Stitch*** pairs Hawaiian girl and alien pet, luscious retro watercolors.

The ChubbChubbs wins AMPAS Award.

The Cathedral is Best Short at SIGGRAPH.

Retro Studios' ***Metroid Prime*** wins GDC Best Game. Digital Illusions' **Battlefield 1942**, Rockstar North's ***Grand Theft Auto: Vice City***, Ubisoft's ***Tom Clancy's Splinter Cell***; EA's ***Medal of Honor: Allied Assault***, Square Soft's ***Kingdom Hearts***; ***Animal Crossing***, Oddworld's ***Munch's Odyssey***.

Nvidia's **Cg** programming language.

Apple Computer buys Nothing Real's **Shake**.

Mill Film and **The Secret Lab**, Disney's VFX group, close.

Massive Software opens in Wellington, N.Z.; **Sony Pictures Animation** and **Luxology** (Modo) in California.

Framestore/CFC's ***Dinotopia*** TV series.

WB releases ***Baby Looney Tunes*** and ***¡Mucha Lucha!*** series.

Early 2000s Timeline of Computer Animation and Visual Effects

Categories

- VFX Movies
- Animated Features
- Independent Shorts
- Computer Games
- Technology / Events
- Related Tech./Events
- Television

2003

Visual Effects Movies:
The Lord of the Rings: The Return of the King *
Master and Commander: The Far Side of the World
Pirates of the Caribbean: The Curse of the Black Pearl

2 Fast 2 Furious
Bad Boys 2
Charlie's Angels: Full Throttle
Daredevil
The Haunted Mansion
Hulk / *The Italian Job*
Kangaroo Jack
Lara Croft: The Cradle of Life
The Last Samurai
The League of Extraordinary Gentlemen
Looney Tunes: Back in Action
The Matrix: Reloaded
The Matrix: Revolutions
Spy Kids 3: Game Over
Terminator 3: Rise of the Machines
X2: X-Men United

The Visual Effects Society holds its first awards competition, www.ves.org. ***The Lord of the Rings: The Two Towers*** wins in multiple categories.

BioWare's ***Star Wars: Knights of the Old Republic*** wins GDC Best Game award.

Infinity Ward's ***Call of Duty***, Ubisoft's ***Prince of Persia: The Sands of Time*** and ***Uru: Ages Beyond Myst***, Nintendo EAD's ***The Legend of Zelda: The Wind Waker***

Nokia's **N-Gage** combines a mobile telephone and handheld game system.

Sony's ***HDCAM-SR*** format for HD video.

2003 (cont.)

Animated Features:
Finding Nemo *
Brother Bear
Les Triplettes de Belleville

The Animatrix (DTV)
Bionicle: Mask of Light (DTV)
El Cid, The Legend AKA *El Cid: La leyenda*
Jester Till AKA *Till Eulenspiegel*
The Jungle Book 2
Kaena: The Prophecy AKA *Kaena: La prophétie*
Millenium Actress AKA *Sennen joyû*
Piglet's Big Movie
Pokemon Heroes
The Rain Children AKA *Les Enfants de la pluie*
Raining Cats and Frogs AKA *La Prophétie des grenouilles*
Rugrats Go Wild!
Sinbad: Legend of the Seven Seas
Sky Blue AKA *Wonderful Days*
Tokyo Godfathers

Pixar's visually spectacular underwater adventure ***Finding Nemo*** leads animated features. Korean ***Sky Blue*** has dazzling looks but opaque storytelling. Some computer animation in ***Les Triplettes de Belleville***, a traditional production with original style.

Harvie Krumpet by Adam Elliot wins AMPAS Short Award.

Pixar's ***Boundin'***, Disney's ***Destino*** (inspired by Salvador Dali's designs), and Blue Sky's ***Gone Nutty***.

WB releases ***Xiaolin Showdown*** and ***Star Wars: Clone Wars*** TV series.

2004

Visual Effects Movies:
Spider-Man 2 *
Harry Potter and the Prisoner of Azkaban
I, Robot

Around the World in 80 Days
The Aviator
AVP: Alien vs. Predator
Blade: Trinity
Catwoman
The Chronicles of Riddick
The Day After Tomorrow
Eternal Sunshine of the Spotless Mind
Hellboy
House of Flying Daggers AKA *hi mian mai fu*
Kill Bill-Vol. 2
Kung Fu Hustle AKA *Kung fu*
Lemony Snicket's A Series of Unfortunate Events
National Treasure
Night Watch AKA *Nochnoy dozor*
The Passion of the Christ
Renegade AKA *Blueberry*
Resident Evil: Apocalypse
Sky Captain and the World of Tomorrow
Thunderbirds
Troy / *Van Helsing*
The Village

Valve Software's ***Half-Life 2*** wins GDC Best Game Award.

Criterion Games/EA's ***Burnout 3: Takedown***; Rockstar North's ***Grand Theft Auto: San Andreas***; Namco's ***Katamari Damacy***, Blizzard Entertainment's ***World of Warcraft***; Nokia's ***Ashen***

Nintendo releases **Nintendo DS**; Sony's **PlayStation Portable** (PSP) released in Japan.

WB releases ***Justice League Unlimited*** and ***The Batman*** TV series.

2004 (cont.)

Animated Movies:
The Incredibles *
Shark Tale
Shrek 2

Appleseed AKA *Appurushîdo*
Los Balunis (DTV)
Bionicle 2: Legends of Metru Nui (DTV)
Ghost in the Shell 2: Innocence
Hair High
Home on the Range
Howl's Moving Castle AKA *Hauru no ugoku shiro*
Immortal AKA *Immortel (ad vitam)*
Pinocchio 3000
The Polar Express
Popeye's Voyage: The Quest for Pappy (DTV)
The Snurks AKA *Back to Gaya*
Steamboy AKA *Suchîmubôi*
Teacher's Pet
Team America: World Police
Terkel in Trouble AKA *Terkel i knibe*

Chris Landreth's ***Ryan*** wins AMPAS Award.

Also nominated for Best Short: ***Birthday Boy*** by Sejong Park and A. Gregory, Blur's ***Gopher Broke***, ***Guard Dog*** by Bill Plympton, and Disney's ***Lorenzo***.

Panavision's **Genesis** HD camera.

HDV format uses 1/4 in. DV tape developed by JVC and Sony.

Canon **XL-2** prosumer 3-CCD camera

Autodesk buys **Kaydara**, with Motion Builder and FBX file interchange format.

Thomson buys **The Moving Picture Company** (MPC).

2005

Visual Effects Movies:
King Kong *
The Chronicles of Narnia: The Lion, the Witch and the Wardrobe
War of the Worlds

Aeon Flux
Batman Begins
Constantine
Doom
Fantastic Four
Harry Potter and the Goblet of Fire
The Island
Jarhead
Kingdom of Heaven
Land of the Dead
Empire of the Wolves AKA *L'Empire des loups*
Memoirs of a Geisha
Negotiator: Mashita Masayoshi AKA *Kôshônin Mashita Masayoshi*
The Promise AKA *Wu ji*
The Ring Two
Shinobi
Sin City
Star Wars: Episode III-Revenge of the Sith

Microsoft introduces the **Xbox 360**, capable of rendering 500 million triangles per second, and Nintendo releases the **Wii**.

Sony's ***Shadow of the Colossus*** wins Best Game GDC award.

Nintendo EAD's ***Animal Crossing: Wild World***, Sony's ***God of War***, Harmonix Music Systems/RedOctane ***Guitar Hero***; Lionhead Studios/Activision's ***The Movies***

Late 2000s Timeline of Computer Animation and Visual Effects

2005 (cont.)

Animated Features:
***Wallace & Gromit in the Curse of the Were-Rabbit* ***
Howl's Moving Castle
Tim Burton's Corpse Bride

Bionicle 3: Web of Shadows (DTV)
Chicken Little
Dragon Blade
Final Fantasy VII: Advent Children
Gisaku
Hoodwinked!
Madagascar
Midsummer Dream AKA ***El sueño de una noche de San Juan***
Olentzero y el tronco mágico
Pooh's Heffalump Movie
Robots
Thru the Moebius Strip
Valiant

Madagascar makes great use of squash and stretch in computer-animated cartoon characters.

Hoodwinked! features limited animation but also humorous gags, and surprises at box office.

The Moon and the Son: An Imagined Conversation by John Canemaker and P. Stern wins AMPAS Best Short Award.

Also nominated for Best Short: ***9*** by Shane Acker, and Pixar's ***One Man Band***.

Canon **XL-H1** HDV camera records on 1/4 in. videotape.

YouTube launches as a video sharing website.

WB releases ***Loonatics Unleashed*** and ***Krypto the Superdog*** series.

2006

Visual Effects Movies:
***Pirates of the Caribbean: Dead Man's Chest* ***
Poseidon
Superman Returns

Apocalypto
Arthur and the Invisibles AKA ***Arthur et les Minimoys***
Battle of Wits AKA ***Muk gong***
Blood Diamond
Charlotte's Web
Children of Men
Curse of the Golden Flower AKA ***Man cheng jin dai huang jin jia***
The Da Vinci Code
Day Watch AKA ***Dnevnoy dozor***
Eragon
The Fast and the Furious: Tokyo Drift
Final Destination 3
Flags of Our Fathers
The Fountain
The Host AKA ***Gwoemul***
Lady in the Water
Letters from Iwo Jima
Mission: Impossible III
Pan's Labyrinth AKA ***El laberinto del fauno***
Rescue Dawn
The Science of Sleep AKA ***La Science des rêves***
Sinking of Japan AKA ***Nihon chinbotsu***
Snakes on a Plane
Ultraviolet
World Trade Center
X-Men: The Last Stand AKA ***X-Men 3***

The Walt Disney Company buys **Pixar**.

Autodesk buys **Alias**.

Nvidia's free hardware renderer **Gelato**.

Apple Computer switches to **Intel** CPUs.

CPU maker **AMD** merges with graphics card maker **ATI**.

Google buys **YouTube**.

2006 (cont.)

Animated Features:
***Happy Feet* ***
Cars
Monster House

A Scanner Darkly
Ant Bully
Azur & Asmar
The Barnyard
Cristobal Molón
Doogal
Everyone's Hero
Flushed Away
Free Jimmy
Happily N'Ever After
Hui Buh: The Castle Ghost AKA ***Hui Buh - Das Schlossgespenst***
Ice Age: The Meltdown
Impy's Island AKA ***Urmel aus dem Eis***
The Land Before Time XII: The Great Day of the Flyers (DTV)
Open Season
Over the Hedge
Paprika
Renaissance
Tales from Earthsea AKA ***Gedo senki***
Tekkonkinkreet AKA ***Tekon kinkurîto***
The Ugly Duckling and Me! AKA ***Den Grimme ælling og mig***
The Wild

Pixar's short ***Lifted***.

Sony introduces the **PlayStation 3**.

Epic Games' ***Gears of War*** wins Best Game GDC award.

Nintendo's ***Wii Sports***; Clover Studio/Capcom's ***Okami***; Bethesda Game Studios/2K Games' ***The Elder Scrolls IV: Oblivion***; Nintendo EAD's ***The Legend of Zelda: Twilight Princess***; Ubisoft's ***Rayman Raving Rabbids***

Warner Bros. releases ***Tom and Jerry Tales*** and ***Legion of Super Heroes*** series.

2007

Visual Effects Movies:
The Golden Compass*
Pirates of the Caribbean: At World's End
Transformers

28 Weeks Later
30 Days of Night
300
4: Rise of the Silver Surfer
Alvin and the Chipmunks
AVPR: Aliens vs Predator–Requiem
Blades of Glory
The Bourne Ultimatum
Bridge to Terabithia
Enchanted
Evan Almighty
Ghost Rider
I Am Legend
The Kite Runner
Live Free or Die Hard
Ocean's Thirteen
Resident Evil: Extinction AKA ***Resident Evil 3***
Spider-Man 3
Sunshine
Sweeney Todd: the Demon Barber of Fleet Street
There Will Be Blood
TMNT AKA ***Teenage Mutant Ninja Turtles: Immortal***
We Own the Night
The Water Horse: Legend of the Deep
Zodiac

Valve's ***Portal*** wins GDC Best Game award.

Ubisoft's ***Assassin's Creed***; 2K Games' ***BioShock***; Crytek/EA's ***Crysis***; Infinity Ward/Activision's ***Call of Duty 4: Modern Warfare***; Valve's ***Half-Life 2: Episode 2***; Harmonix/MTV Games' ***Rock Band***; Nintendo EAD's ***Super Mario Galaxy***

Nokia's **N-Gage** gaming capabilities available to mobile phones.

2007 (cont.)

Animated Features:
***Ratatouille* ***
Persepolis
Surf's Up

Bee Movie
Beowulf
Donkey Xote
Appleseed Saga: Ex Machina AKA ***Ekusu makina***
Evangelion: 1.0 You Are (Not) Alone AKA ***Evangerion shin gekijôban: Jo***
Fear(s) of the Dark AKA ***Peur(s) du noir***
Jungo Goes Bananas AKA ***Jungledyret Hugo: Fræk, flabet og fri***
The Land Before Time XIII: The Wisdom of Friends (DTV)
Lissi and the Wild Emperor AKA ***Lissi und der wilde Kaiser***
Meet the Robinsons
Nocturna
The Secret of the Magic Gourd AKA ***Bao hu lu de mi mi***
Shrek the Third
The Simpsons Movie
Vexille AKA ***Bekushiru: 2077 Nihon sakoku***

BUF Compagnie's ***Même les pigeons vont au paradis (Even Pigeons Go to Heaven)***

Oktapodi is Best of Show at SIGGRAPH.

The **Red One** HD camera is released.

Dalsa introduces the **Origin II** camera.

Nvidia buys **Mental Images**.

Autodesk buys **Skymatter**'s **Mudbox**.

Late 2000s Timeline of Computer Animation and Visual Effects

Categories

- VFX Movies
- Animated Features
- Independent Shorts
- Computer Games
- Technology / Events
- Related Tech./Events
- Television

2008

Visual Effects Movies:
The Curious Case of Benjamin Button *
The Dark Knight
Iron Man

Asterix in the Olympic Games AKA *Astérix aux jeux olympiques*
Australia
Babylon A.D.
Bedtime Stories
City of Ember
The Chronicles of Narnia: Prince Caspian
Cloverfield
The Day the Earth Stood Still
Hancock
Harry Potter and the Half-Blood Prince
Hellboy II: The Golden Army
The Incredible Hulk
Indiana Jones and the Kingdom of the Crystal SkullJourney to the Center of the Earth
Jumper
The Kingdom and the Beauty AKA *Kwong saan mei yan*
The Machine Girl AKA *Kataude mashin gâru*
The Mummy: Tomb of the Dragon Emperor
Pineapple Express
Quantum of Solace
Speed Racer
The Spiderwick Chronicles
The Spirit
Starship Troopers 3: Marauder (DTV)
Tropic Thunder
Vantage Point
Wanted
The X Files: I Want to Believe
You Don't Mess with the Zohan

Autodesk buys **Realviz** and **Softimage**.

Vivendi buys **Activision**.

2008 (cont.)

Animated Features:
Wall•E *
Bolt
Kung Fu Panda

$9.99
Agent Crush
The Clone Wars
Delgo
Dr. Seuss' Horton Hears a Who!
Dragon Hunters AKA *Chasseurs de dragons*
Fly Me to the Moon
Foodfight!
Ghatothkach: Master of Magic
Goat Story: The Old Prague Legends AKA *Kozí příběh–Pověsti staré Prahy*
Igor
Idiots and Angels
Journey to Saturn AKA *Rejsen til Saturn*
Madagascar: Escape 2 Africa
Mia and the Migoo AKA *Mia et le Migou*
Niko & the Way to the Stars AKA *Niko - Lentäjän poika*
Open Season 2
Ponyo on the Cliff AKA *Gake no ue no Ponyo* (Japan release)
The Sky Crawlers AKA *Sukai kurora*
Space Chimps
The Tale of Despereaux
Waltz with Bashir

Pixar's short *Presto*

Blu-ray Disc becomes the standard for high-capacity DVD.

Sony Pictures releases **Hancock** feature movie in 4K digital projection.

Cartoon Network releases *Star Wars: The Clone Wars*, a 3D CGI series produced by Lucasfilm Animation.

WB's *Batman: The Brave and the Bold*

2008/2009

Square Enix's *Chrono Trigger* (DS); Square Enix's *Crisis Core: Final Fantasy VII* (PSP); Lionhead Studios' *Fable II* (Xbox 360); Bethesda Game Studios' *Fallout 3* (Xbox 360/PS3/PC); Ubisoft's *Far Cry 2* (Xbox 360/PS3/PC); Epic's *Gears of War 2* (Xbox 360); Rockstar North's *Grand Theft Auto IV* (Xbox 360/PS3/PC); Mistwalker's *Lost Odyssey* (Xbox 360); Konami's *Metal Gear Solid 4* (PS3); Capcom's *Monster Hunter* (PSP); Nokia's *One*; Q Games' *Pixel Junk Eden* (PS3); Black Rock's *Pure* (Xbox 360/PS3/PC); Sony SCE's *Siren: Blood Curse* (PS3); Maxis' *Spore* (PC/Apple); Ironclad Games' *Sins of a Solar Empire* (PC); Nintendo's *Advance Wars: Days of Ruin* (DS); Sega's *Yakuza 2* (PS2)

(2009)

Visual Effects Movies Scheduled and/or in Production:
2012
Angels & Demons
Arthur et la vengeance de Maltazard
Avatar
Enter the Void
G.I. Joe: The Rise of Cobra
Halo
The Lovely Bones
Shanghai / *Sin City 2*
Transformers: Revenge of the Fallen
Star Trek
Terminator Salvation
Where the Wild Things Are
X-Men Origins: Wolverine
The Wolfman

2009/2010

Animated Features Scheduled and/or in Production:
9
1906
A Monster in Paris AKA *Un monstre à Paris*
Astro Boy
Chico & Rita
A Christmas Carol
Cloudy with a Chance of Meatballs
Coraline
Evangelion: 2.0 You Can (Not) Advance AKA *Evangerion shin gekijôban: Ha*
Ice Age: Dawn of the Dinosaurs AKA *Ice Age 3*
The Legend of Spyro
Monsters vs. Aliens
Neanderthals
Open Season 2
Planet 51
The Princess and the Frog
Up

(2010)

Visual Effects Movies Scheduled and/or in Production:
Alice in Wonderland
Arthur et la guerre des deux mondes
The Chronicles of Narnia: The Voyage of the Dawn Treader
Gears of War
The Green Hornet
Harry Potter and the Deathly Hallows I
Iron Man II
Jonny Quest
Jurassic Park IV
Kung Fu Hustle 2
Lincoln
Logan's Run
Prince of Persia: The Sands of Time
The Smurfs / *Thor*
When Worlds Collide
xXx: The Return of Xander Cage

2010/2012

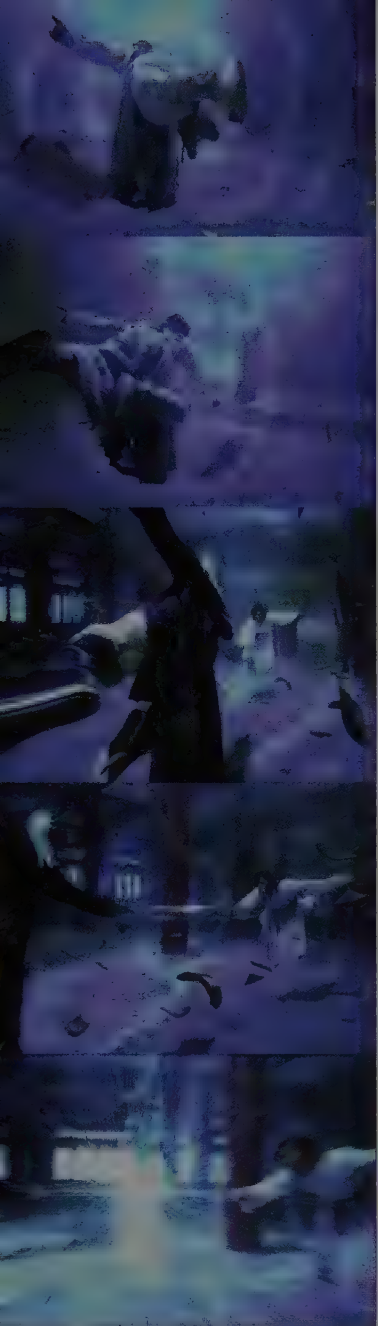
Animated Features Scheduled and/or in Production:
The Bear and the Bow
Cars 2
Evangelion 3
Gnomeo and Juliet
Guardians of Ga'hoole
Hoodwinked 2: Hood vs. Evil
How to Train Your Dragon
John Carter of Mars
Rapunzel
Shrek Goes Fourth
Tim Burton's Alice In Wonderland
The Tinker Bell: A Midsummer Storm
Tintin
Toy Story 3

(2011/2012)

The Avengers
Bond 23
Harry Potter and the Deathly Hallows II
Interstellar
Justice League: Mortal
Pirates of the Caribbean 4
Spiderman 4
Superman: Man of Steel
TR2N

(2011/2012)

King of the Elves
Kung Fu Panda 2
Madagascar 3
Newt
The Three Musketeers
Tusker



1.6.2 Affordable technology and qualified professionals have spawned many new VFX companies. (*Duel* courtesy of Menfond. © China Star Entertainment Ltd./ Win's Entertainment Limited.)

CHAPTER 1

Key Terms

Academy of Motion Picture Arts and Sciences (AMPAS)
CADAM
CAPS
Cinefex
Codecs
Computer-aided design and manufacturing
Computer intranets
Computer graphics technology
Computer games
D-20, Arriflex camera
D-Cinema, Digital cinema
Digital environment
Digital imaging techniques
Digital information
Digital movie projectors
Digital video
Dreamcast
GameCube
Gelato
Genesis effect
Geometry Engine
F950 CineAlta camera
High definition digital video
Kinetophones
Kinetoscope
LBE rides
Linux
Location-based entertainment
Mac OSX
Mainframes
Massively parallel computers
Microcomputers
Minicomputers
Motion Picture Association of America (MPAA)
Music videos
Nintendo 64, Wii
Phantom camera
Platform games
PlayStation
PlayStation 2, 3
Red One camera
RenderMan
Scanimate
Seventh art
SIGGRAPH
SIGGRAPH Video Review
Silicon chip
Sketchpad
Supermicrocomputers
Supercomputers
Synthavision
Time-sharing
UNIX
Viper FilmStream camera
Windows NT
Workstations
www.artof3d.com
Xbox, Xbox 360

(Image courtesy of Nvidia Corp.)

CHAPTER 2

Creative Development and the Digital Process

QUICK INDEX

(© Harald Siepermann.)

Summary

THERE ARE MANY DIFFERENT WAYS TO DEVELOP, design, and produce a sequence of three-dimensional computer animation or a visual effects shot. There are as many production methods as there are different types of projects with different resource allocations and different creative goals. This chapter presents several methodologies for digital production and explains what makes each one of them unique and what each is best suited for.

2.1 Storytelling

Stories are the most common and most powerful vehicle we use to talk about life. Not just one life but many lives. Life in general, and our own lives in particular. Past, present, and future lives. Real, imagined, and assumed lives. Inspiring, intriguing, tormented, or impossible lives.

Stories communicate facts. Stories provide answers to questions. Stories make us feel different emotions. Stories sometimes even provoke actions that shape reality. Whether they are linear or nonlinear, whether they depict an event with cartoon characters or a colorful dance of abstract shapes, stories are the essence of animation. Being a good storyteller requires many talents and skills. But why should anybody interested in computer animation have to learn about storytelling? Because animations are more than just moving images. Animations tell stories and communicate emotions that are initially drafted in screenplays, and later in storyboards and character sheets.

In most cases, the work of animators involves the **visual interpretation** of a story and its characters (Figs. 2.1.1 and 2.1.3). Animators translate the personality of characters into facial expressions, gestures, and motions, whether the story is a complex epic drama between nations or the simple courting of a lady sphere by a male square. Animators and other visual people involved in the production of animations, often start to sketch their visual interpretations of a story by reading and discussing a screenplay.

2.1.1 Characters work best within the story when their design, look, and behavior is appropriate to the type of story and style of storytelling. (*How to drive everybody crazy*, © 2008 TeamTO–France 3–Cake Entertainment.)

The power of storytelling comes from the fact that good stories entertain us while we learn something about life. When putting stories together one must consider the audience, as crucial aspects of the story—such as topic, point of view, and treatment—are driven by the audience. Good storytelling engages the audience's imagination and it does so not with dry explanations but by expressing emotion—this is called dramatization. Audiences empathize with emotion.

Story Development and Scriptwriting

Behind any great visual project there is a strong idea and a great story. The stage of **story development** takes place before production begins, especially in small projects where the writer develops the story before any other preproduction has started or any budget has been committed to the project. In larger projects story development marks the beginning of development. During this process the writer or game designer (or story/design team in a larger project) has to focus on developing the characters and the plot, but also has to adapt it to the strengths and limitations of the visual medium in question. Usually a few creative cycles and/or revisions exist between a story or a game idea and the final **script** or the game design document.

In addition to their acting and puppeteering skills, some animators have stories to tell. As you set to write a story, consider what genre of stories you are good at telling and what topics you believe in. Writers write better stories and use less clichés when they write about topics that they believe in. In general terms, stories can be classified in categories or genres. Each **story genre** has its own characteristics, and some of the popular genres include comedy, romantic, action/adventure, fantasy, science fiction, tragedy, horror, and drama.

A **screenplay** is a written document that tells a story by using descriptions, dialog, and some production notes. Unlike a novel, which is written to be printed and read, a screenplay is not an end-product in itself. Instead, the screenplay is an intermediate work, a vehicle for the story to be retold with images in the form of an animation, a movie, or a play. For this reason, screenplays tell stories in ways that can be **translated into moving images**. When written in the proper format, one screenplay page is usually equivalent to one minute of action on the screen. Screenplays can differ by the amount of dialog they incorporate, the number of characters they present, or the detail of their descriptions of imagery. But what all screenplays have in common is a subject and a clearly defined treatment of the subject that is adequate for the intended audience.

The **treatment of the subject** in a screenplay is defined by the point of view that the storyteller wants to convey to the audience. Treatments can be, for example, dramatic, comic, or lyrical, action-packed, or introspective. Considering the **intended audience** can make it easier to define the treatment for a screenplay's subject. This includes not only the philosophical or political treatments of the subject, for example, but also the visual treatment. This concerns anima-

Storytelling Tips

1. Write stories and use storytelling styles that are interesting to your intended audience.
2. Keep the interest of the audience by engaging their imagination and empathy.
3. Pause, to let your characters, and the audience, think.
4. Surprise your audience by keeping them ahead and behind of the story, by revealing at the right moment.
5. Use a dramatic arc to show that the characters changed.
6. Use parallel action and flashbacks to add temporal dimensions to your story.
7. Use the principles of animation to enhance your storytelling.
8. Save the best for last; the climax must be most satisfying.

2.1.2 Tips to improve your storytelling.

2.1.3 The artistry of *Kung Fu Panda* exemplifies storytelling that is enhanced by the visual richness of the animated scenes and the relation of the visual style to the story. (*Kung Fu Panda*™ and © 2008 DreamWorks Animation LLC. Used with permission.)

tions because some screenplays may require very simple or specific computer animation techniques to achieve the desired effect or emotion. The **subject of a screenplay** is defined by what the story is about, who the characters are, and what happens to them throughout the sequence of events. Because the subject of a story is often presented through the actions of a character, it is important to develop the personality of all the characters before the story is told.

Basic Story Structure and Storytelling Techniques

Stories can be designed and told in many ways. Some story designs are easier to digest by a mainstream audience than others. Consider a **classical story design** if you are interested in reaching mainstream audiences. Try enhancing your own story ideas by incorporating the storytelling techniques developed throughout the centuries, as you also experiment with new ways to tell stories. Explore the classic comedies and dramas, as well as short stories, stage screenplays, film scripts, and poetry. A classical story design includes a single active protagonist facing external conflict and/or an antagonist (or both) within a consistent reality and with a closed ending, among other things. In designing your story choose the moments and events that matter in the narrative of the character. Meaningful moments are about meaningful change. Meaning produces emotion. Change occurs through conflict, and the resolution of conflict. Consider the topic and setting of your story: the driving challenge (what?), the motivations and changes (why?), the period (when?), the duration (how long?), the location (where?), and the level of challenge or conflict.

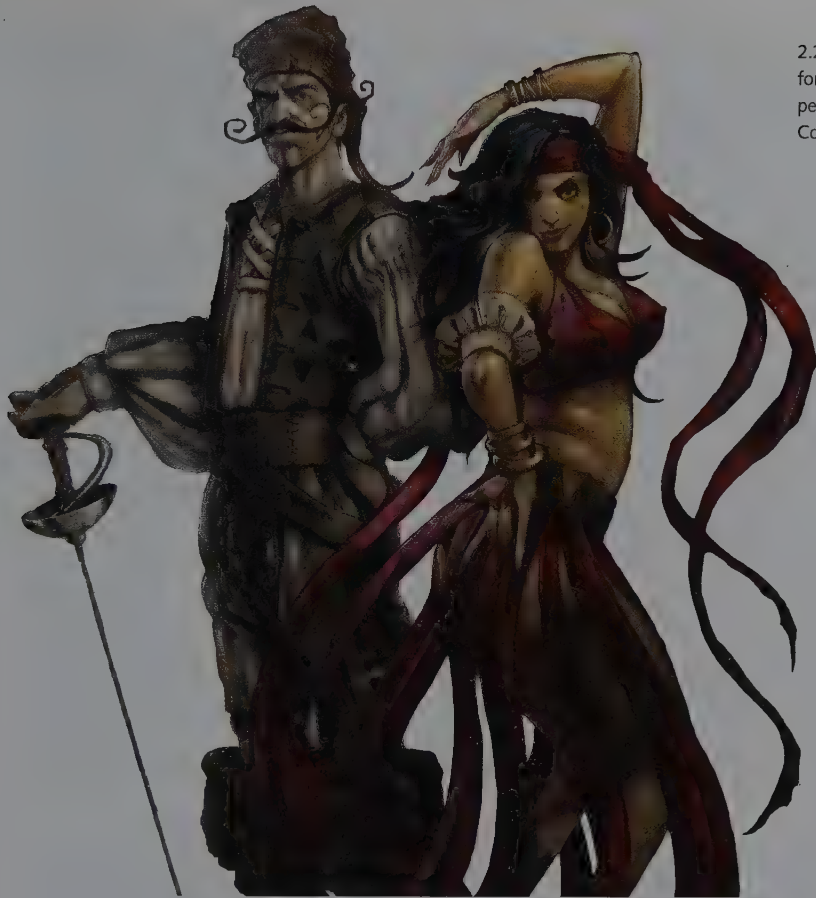
(*Dragon Hunters* © MMVII Futurikon Films, Trixter, LuxAnimation, France3 Cinéma, RTL-Tvi, in coproduction with Mac Guff Ligne.)

A screenplay's subject can be presented in the context of a variety of dramatic and narrative structures. The **structure** of a story holds all the parts of the story together. There are no rigid rules about the best way to structure a story, and stories can be told with a wide variety of styles and techniques. As agreed by a variety of storytelling traditions, stories that are told in the context of a **linear** medium—such as film or video—consist of a beginning, a middle, and an end, and the action in a story usually moves from one event to another. A classical story design in the European tradition—for example, Greek and Shakespearean dramas—has five ingredients: an inciting incident, progressive complications, a crisis, a climax, and a resolution. These ingredients usually take place in the story in the order listed but the sequence may be creatively modified while preserving the overall effect. The narrative structure of classical Asian narratives is not identical but somewhat similar to its European counterpart. In Chinese and Japanese kishõtengõ (起承転合) or kishõtenketsu (起承転結), for example, a four-part narrative structure is prescribed: introduction, development, twist, and conclusion.

In the most traditional sense, the **beginning of the story**—also called exposition or setup—usually introduces the main characters, establishes the dramatic premise, and sets up the events and situations that will develop the story. The **middle of the story**—also called confrontation or climax—usually contains the moments when the main characters confront the conflicts that when resolved will lead to a resolution. The **end of the story**—also called resolution or *dénouement,* from the French "untying a knot"—usually contains the outcome of the dramatic sequence of events in the story.

2.2.1 (Above and below) Character designs rendered with ink and watercolor. (© 2002 Harald Siepermann.)

The **inciting incident** is a catalyst event that brings conflict to the life of the main character. In classically structured stories the inciting incident is usually placed at the end of the initial **exposition**, where the background information necessary to understand the story is presented. Classical characters actively try to restore the balance that was upset by the inciting incident, sometimes achieving it or not, or only partially. The **progressive complications** are the situations and events that create **rising action** from the moment of the inciting incident all the way through the climax of the story. These complications take the characters through the conflicts at hand, giving them the opportunity to negotiate turning points and go through emotional transitions. A feature-length animated movie offers more opportunity



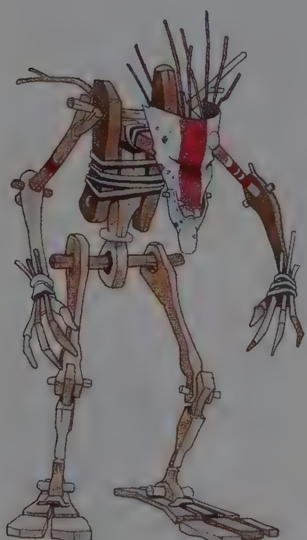
2.2.2 Two conceptual character designs for the *Fable II* game. (Reprinted with permission from Microsoft Corporation.)

for depth, complication, and contradiction than a short film. The former usually spreads the progressive complications throughout three acts, while the latter may be limited to a single-act plot; classical plays from antiquity often had five acts. The **crisis** is a powerful, usually emotional, moment of raising action where the main character is confronted with a situation and an ultimate choice. The handling of the crisis by the main character has a definitive impact over the course of events. The **climax** is the biggest turning point and the moment of most significant change in the story: a moment of absolute change. All characters in the story, but especially the protagonist, are deeply and irreversibly impacted by the outcomes of the climax. After the climax comes the **resolution**, where everything is revealed and comes to a conclusion, where conflicts are resolved and the fortunes of characters move in a positive direction—for example, *they lived happily ever after*—or a negative one—for example, failure, misfortune, or catastrophe.

Several variations of the traditional structure of a story are possible. For example, the structure can be altered so that the story starts with the resolution, continues with the setup, and concludes with the confrontation. But in every instance stories are always told

2.2.3 Character design for the *Rayman* game. (© Ubisoft Entertainment. All rights reserved. Rayman, Rayman Raving Rabbids, the character of Rayman, Ubisoft and the Ubisoft logo are trademarks of Ubisoft Entertainment in the U.S. and/or other countries.)

2.2.4 Concept painting indicating the shape, coloring, texture, and resting pose of a character. (© 1999 Oddworld Inhabitants, Inc. All rights reserved.)



2.2.5 Colored line sketch of shape and shading style in *Suba Jellyfish*. (© 2005 Alastair Graham.)

2.2.6 (Opposite page) The look of the main characters in *Shrek* and *Monsters, Inc.* was developed through an iterative process of intuition, design, creativity, selection, critique, and refinement. (Top: Shrek™ and © 2001 DreamWorks L.L.C. Bottom: © Disney Enterprises, Inc./Pixar Animation Studios.)

in terms of **events** or **plot points** that make the story evolve in a particular way. These events are the moments in the story when the action takes a different turn. These moments and events help develop the story and keep the action moving.

There is a long road to travel between coming up with a good story idea and crafting a well-told story. Keep in mind the storytelling tips listed in Figure 2.1.2. The importance of pauses in narration, dialog, and action cannot be overemphasized, as **pausing** represents the thinking of the character and it also gives audiences the opportunity to think. A balance between **suspense** and **surprise** will keep your audience ahead of the story and behind the story. Reveal only at the right moment. **Parallel action** adds crucial temporal dimensions to your story.

The telling of a story with motion pictures, and more specifically with animation, can be greatly enhanced by using the natural strengths of the medium. In addition to a solid story structure, compelling characters, deep characterization, and engaging dialog, use the animation-specific storytelling techniques at your disposal. These techniques include the principles of animation, particularly timing (an acting concept), physical acting, pantomime (especially in the absence of dialog), facial expressions, visual humor, cinematic staging of actions, cinematic lighting and camera moves, music soundtrack, effects animation, and image sequencing.

Nonlinear Storytelling

Storytelling in most visual media happens in a linear way because images, sounds, and text follow each other in a single predetermined order. But a variety of possible sequences may be possible in an **interactive project** because users can make different requests and also follow a variety of paths. Interactive projects may have multiple endings and even multiple beginnings each time they are played. Storytelling for interactive media requires unique techniques because of the **nonlinearity** of the media.

Creators of an interactive project have in **flowcharts** a powerful tool for planning the project and for determining the many paths that the story may follow. Flowcharts are diagrams that clearly lay out all the **branching** options that may occur in the **flow of events** of interactive dialog. The branching structure in an interactive system may be simple if few options are offered or complex if the options are multiple. Each branching node in a flowchart is controlled by a choice made by the individual or individuals interacting with the system. When a choice is made at a branching node the flow of events advances to another **hierarchy level** in the flowchart. On occasion, the sequencing of events in an interactive project may be sketched out in the form of a traditional storyboard. The **interactivity** of a computer system is based on the **dialog** established between the system itself and the individuals using it. Individuals using interactive systems can make their choices through standard

2.3.1 Concept sketch of an environment (top right) created during the visual development stage of production. Color keys (above) help to define the palette and mood of a particular shot. (© 1999 Oddworld Inhabitants, Inc. Oddworld Inhabitants and the Oddworld Logo are Registered Trademarks of Oddworld Inhabitants, Inc. All rights reserved.)

2.3.2 (Opposite page) Concepts for a sky stadium and a shark tank. (Top: Copyright by Tino Schaedler. Bottom: Copyright by Tino Schaedler and Daniel Widrig.)

input peripherals such as mice, joysticks, and keyboards, or unique ones such as gloves and bodysuits with ultrasonic and light sensors that determine the position, orientation, and physical gestures of a person. Interactive media systems are usually built around a computer system that is able to control the flow of information stored in a variety of media, formats, and systems, including still and moving images, sound, and text. Hence the name interactive multimedia.

Flowcharts describe in an abstract manner the overall structure and dynamics of an interactive project. But **scripts** address the flow of events in an interactive project; they are the practical implementation of the ideas contained in a flowchart. Scripts are computer programs that collect and evaluate information about the choices made by the system's users, and then direct the program in the appropriate direction. Scripts trigger events that may include displaying an image or playing a sequence of images or sounds.

Interactive media are not made to be just watched or read; instead, they are made to be used by people. Three-dimensional real-time computer games are made to be played. For this reason, the functionality of an interactive system should always be checked with extensive **user testing**. The feedback and suggestions of users usually uncover the moments in the flow of events that may be confusing, or important functionalities that may be missing, or users' requests that crash the program and freeze the system. Only after a thorough user testing process can interactive nonlinear storytelling projects be released to the public.

Users of interactive narratives are now able to directly add characters, environments and storylines, particularly in the realm of **massively multiplayer online games**, also called **MMOG**s. User-created content adds new dimensions to nonlinear storytelling (Fig. 4.7.2).

2.3.3 Production paintings help to visualize the director's vision. This artwork by Michael Meier for Roland Emmerich's *10,000 BC* was sketched with pencils, then scanned and painted digitally with Corel Painter and Photoshop. (© Warner Bros. Entertainment Inc. All rights reserved.)

2.2 Character Design

After the story, the characters are perhaps the second most important aspect of an animated project (the third would be the quality of the animation itself). One of Disney's twelve principles highlights this fact by emphasizing that characters must have *appeal*. As illustrated in Figures 2.2.1–2.2.6 the look and personality of characters is developed through drawings, sculptures, maquettes, and even computer-generated renderings. Visually speaking, it is important to determine early on the type of look for the character in question: cartoony, stylized, or realistic (see Chapter 10 for more details on character development). Character sheets and character turnarounds are two important deliverables from this stage of preproduction (Figs. 10.4.7 and 10.5.1). The former consist of sets of drawings that define the attitudes and poses of the characters in the form of body positions and facial expressions, and the latter show the key features of a character from different points of view.

In developing the look of characters for computer animation one must keep in mind which techniques will be used to animate it and deliver it to an audience. A character for a real-time action computer game, for example, would have to be designed very differently from a character in a feature animated film. A secondary character in an all-computer-animated film would be designed differently from the starring virtual actor in a visual effects movie. **Props** are used in a scene ranging from a small utensil to a vehicle (Figs. 6.7.4, 6.10.9, 9.2.3, and 9.3.14), and are often designed by assistant character designers or prop specialists and individuals with industrial design backgrounds. It is of critical importance for the character designers to consult with the technical supervisors of modeling, setting up the animation controls, and rendering the character. A two-way open communication between the creative and technical teams regarding functionalities and limitations only improves the full realization of the creative vision.

(ReBoot® and © 1997 Mainframe Entertainment, Inc. All rights reserved.)

2.3 Visual and Look Development

Visual development is about setting a visual direction and a **visual look** for the project (Figs. 2.3.1–2.3.5). In a computer animated project this stage includes the overall styling and the visual atmosphere; it sometimes includes the types of environments and props, the color schemes for the project, and the integration of the characters into the overall style. The visual development team usually includes painters, sculptors, and illustrators, and it is most active during the preproduction stage. This team is charged with creating **concept art**, as opposed to production art. Concept art is useful to illustrate crucial moments in the storyline, which is important to pitch a project to potential partners, producers, and distributors. Concept art also helps to visualize the vision of the director, illustrating his/her ideas and also proposing additional ideas for consideration. Concept art also helps to sketch out the scope and some of the challenges that the production crews might face. In fact, concept art is oftentimes used to create feasibility tests and proofs of concept. A drawing or painting created with two-dimensional techniques might be used as a guide to recreate the same concept in three dimensions. Figure 2.3.4, for example, shows watercolor paintings used as a guide to define the approach for modeling environments. Figure 2.3.5 shows final production art, meaning still frames from the computer animated movie, that were inspired by the fantastic floating worlds developed during the early visual development stage.

Creating color palettes and a few **color keys** for key scenes in the project is also an important aspect of the visual development process because it sets much of the visual mood of the project (Fig. 2.3.1). The color keys must depict a mood and visually complement the storytelling and emotional purpose of the story. Imagine the difference in mood between a shot that has harmonious soft colors, and the same shot with loud contrasted colors.

(*Dragon Hunters* © MMVII Futurikon Films, Trixter, LuxAnimation, France3 Cinéma, RTL-Tvi, in coproduction with Mac Guff Ligne.)

2.3.4 Monochrome renderings are used to preview the geometry specified for the sets of *Angelina Ballerina* in the concept paintings. (© 2009 Helen Craig Ltd. and Katharine Holabird. By permission of HIT Entertainment Limited.)

2.4 Production Strategies

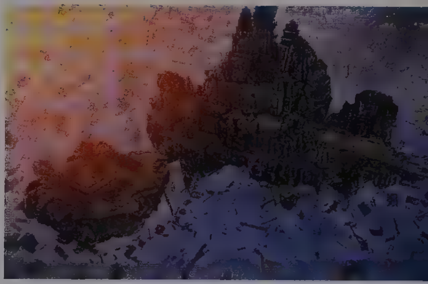
Planning the computer animation **production strategy** for any movie, game, or visual effects project starts with a review of the type of production, the technical complexity, and the basic resources, such as budget, schedule, personnel, and computer systems. Production strategies will vary sometimes radically depending on the unique combination of these factors. Production trends also change from time to time and place to place. In years of economic growth, production budgets are usually generous, but in other years budgets are tight as clients and producers look for savings everywhere. Every once in a while companies and studios set guidelines to lower production costs, and that always has an effect on production.

Types of Production

Computer animation projects can differ a lot from each other depending on what **type of production** they are. An experimental animation short, a visual effects shot for a live action film, a commercial production, an episode for an animated series for television, a real-time platform game, and a feature movie production all have very different purposes and project dynamics. Figure 2.4.1 lists some of the categories of computer animation projects that are common today. Think of the differences that exist between the two extremes: the production of a one-person experimental computer animation and the production of a computer animated feature movie. An experimental short often seeks to explore techniques, topics, or treatments that are not commonly used in mainstream productions. Most experimental works are not developed to be first and foremost crowd pleasers so the director, who often is also the screenwriter and animator, has great freedom to experiment even though production resources in experimental production are usually limited. This fact may have a limiting impact on the technical sophistication and amount of computer systems and personnel that may be employed, but it also may have a positive effect in the form of a flexible production schedule and creative freedom.

Games require **non-linear animation** that can be played in a variety of combinations and for varying amounts of time. Linear computer animations, on the other hand, once completed have a wide range of durations. There is no **standard length** of time that determines what is a short or a long piece of computer animation. Animated shorts usually range between one and ten minutes. Computer animations under one minute are commonly created for experimental pieces, student projects, TV commercials, TV station identification sequences or program openings, and movie titles. Computer animated productions that fall in the medium-length category include episodes for animated series, a collection of visual effects to be inserted in a feature film or TV series, sequences for an amusement park motion ride, scientific visualizations, or architectural walk-

throughs. The standard length of a feature movie is 90 minutes, but for individual or small team productions anything over 30 minutes is long and requires a huge amount of work, time, and energy.



2.3.5 The fantastic *Dragon Hunters* environments recreate the magic and melancholy specified during the visual development process. (© MMVII Futurikon Films, Trixter, LuxAnimation, France3 Cinéma, RTL-Tvi, in coproduction with Mac Guff Ligne.)

Technical Complexity and Delivery Media

The technical complexity of experimental computer animations varies greatly from project to project, but in all cases, the technical challenges in a computer animation have to be chosen carefully so that the production may be completed successfully with the available resources. A feature computer animated movie is usually produced to tell a story that may be appreciated by a particular type of audience, or a mainstream audience. Most productions of this type are quite complex and require a careful plan even when the resources are significant. This allows for ambitious creative and technical challenges, but it also implies great pressure to deliver a product that will sell and will generate returns to partners and investors, and possibly bonuses for the employees.

The **technical complexity** of a computer animation may range from the very simple to the extremely complex. A computer animation project that is technically simple may involve a few objects that are animated with just a single or a few simple motion techniques. Animation for a platform game must take into account and cater to the requirements of the **game engine**, for example, that will play-

3D Computer Animation Sample Projects

- 90-minute computer animated feature movie
- 30 minutes to complement a hand-drawn feature movie
- 15-minute computer animated episode in a weekly series
- 10 minutes of cut-sequences for a role-playing computer game
- 4-minute independent movie
- 10 minutes of visual effects shots for a live action movie
- 2-minute weekly low-resolution animation for online streaming
- 30-second interstitial of motion graphics for a weekly TV show
- 15 seconds of CG visual effects for a live action TV commercial
- 300 moves for a real-time fighting computer game

2.4.1 Some of the most common types of three-dimensional computer animation projects.

2.4.2 The technical complexity of a three-dimensional character drawn in two-dimensions needs to be assessed early in the production process. (© 2002 Harald Siepermann.)

back the animation in real time. A television station identification or news program opener created with three-dimensional animation may be complex from the choreographic point of view but technically simple from the point of view of the motion techniques used. These animations typically involve keyframe animation enhanced with beautifully crafted models and outstanding rendering. A special effects segment produced for a live action feature movie that includes computer animation typically requires several motion techniques such as keyframing, inverse kinematics, and motion dynamics. In this type of project the computer-generated motion has to be perfectly synchronized and aligned with the live action, the live special effects, and the traditional cel animation.

The **delivery medium** or **media** is usually established early in the project since it usually has a significant impact on schedule and budget. It is imperative to clarify the delivery media specifications early on in the project since these specs will determine, among other things, the pixel resolution and the aspect ratio of the images to be rendered (see Chapter 15 for output considerations).

Meetings, Meetings, Meetings...

The best way to deal with technical complexity is by planning thoroughly before production starts. Preproduction planning starts with **meetings** usually with the producers or directors of the project. The goal of these initial meetings is to gather project information related to the technical strategies that may be necessary to generate the desired results. Producers and directors can provide information such as **creative vision**, visual style, budget, and deadlines that affects technical complexity. Effects supervisors, supervising animators, project managers and others also contribute greatly to these early discussions. Once the sequences to be animated are storyboarded it is necessary to meet with the technical directors who are responsible for clarifying complex technical issues, providing technical support, and developing new tools that may be necessary to complete the project. The goal of having these meetings and developing the storyboard is to narrow down the possibilities and to develop a precise **plan of action**. This plan must contain a **technical implementation** for every shot in the computer animation as well as a crystal-clear creative vision that will guide the production. The production plan should also contain a set of **deliverables** including number of frames, complexity of models, and number of effects. The plan also contains **milestones** that help the team set priorities and strategies. Later in the production process meetings become the forum for presenting work in progress and launch the **review** and **approval** cycle.

Planning the Production Workflow

Choosing one technical implementation—or a specific combination of techniques—over all the others usually requires finding a balance

between the best way to achieve the desired result and the least expensive way to do so. This balance should take into account the project as a whole, all of its parts. The technical implementation for each part of the project, for example a single shot in a sequence, starts with an analysis of the elements in the shot and the ways in which they interact with each other. This analysis results in a **written description** of the plan for the shot that covers the elements of the shot, the interactions between them, and the specific techniques that will be used to create them. It is at this stage when the director's vision and the concept art have to be translated into lists of specific deliverables (Fig. 2.4.3).

Designing production workflows is part art and part science. Some of it is based on the analysis of the hard facts like budgets and deadlines. Some of it is based on the careful evaluation of subtler but critical issues like creative goals, personalities of the team members, group dynamics, and the overall production experience of the team. For example, what might work in a large production company might spell disaster in a small studio; what might work in a computer animated character film might not be an option for a visual effects live action film.

Building a successful **production flow** is not done in a vacuum, but by looking at the specifics of the production—and more important—by sharing the proposed flow with the core members of the team, seeking their feedback, and incorporating their feedback into the production plans. In these days of quickly changing production tools, one hears too often about digital productions that miss their goals because of a poorly structured production flow or because of an inflexible producer or production manager who did not listen to the suggestions of more experienced members of the production team. Equally problematic can be a director who is unable to compromise and find a middle ground between the grand vision and what is actually feasible. The responsibility for coming up with the best production flow, also called a production pipeline, for a specific project is usually shared by several individuals involved in the production. This group may include the artistic and technical leads for the project, the assistant producer, the production manager, and other members of the art and animation production teams. It is not uncommon, but far from ideal, for an art director or an assistant producer in a small production to be suckered into doubling as a project manager when a dedicated production manager is not within the budget.

Most experienced production managers or sequence supervisors can structure the best production flow, partly because they have done it before, have made mistakes—and hopefully learned from them—and because they understand the production process. A few experienced production managers, animation directors, or visual effects supervisors are able to intuitively draft a fairly accurate and reasonable production flow just as an experienced *chef de cuisine* is able to cook a delicious dish without having to measure the ingredients or even follow the steps in a recipe.

Breaking Down a Shot or a Digital Production

Creative Issues

- Type of project? (see Fig. 2.1.2)
- Clear creative direction?
- Direction likely to change throughout production?
- Context of shot? What happens before and after?
- Length in seconds or minutes?
- Have to match look of live action? Visual effect or stylized animation?

Technical Issues

- Is client likely to understand technical issues?
- Nature of action?
- Type of motion?
- Complexity of geometry?
- Number of primary and secondary characters? Major and minor elements?
- Type of lens and lighting sources?
- Characteristics of final output medium and process?

Logistical Issues

- Budget amount?
- When is the delivery deadline?
- When are the preliminary review deadlines? Likely to change?
- Size and type of team?
- What animation tools and computing power are available?
- Other materials required before project can start or finish? Likely those will be on time?

2.4.3 Breaking down a visual effects shot or an animation sequence is one of the most important moments during preproduction. This process starts by considering the main creative goals of the project, as well as the major variables that might impact production. The goal of breaking down a shot or project is to end up with specific production tactics that will accomplish the best results within the limitations.

Components of a 3D Computer Animation Studio

- Vision
- Business plan and funds
- Clients and projects
- Space
- Creative personnel
- Production personnel
- Technical personnel
- Administrative and sales personnel
- Turnkey application software
- Proprietary software
- Processing power
- Peripheral storage
- Computer network and communications
- Input and output equipment

2.5.1 Basic components of a 3D computer animation facility.

Types of Software

- 3D surface modeling
- 3D rigging and animation
- 2D/3D scanning
- 3D paint
- Motion capture
- Shading and rendering
- Compositing
- Editing
- Color calibration
- File compression
- Rotoscoping
- Web access
- 2D animation
- 2D paint and retouching
- Media asset management
- Data backup
- Network manager

2.5.2 Common types of applications software found in computer animation and visual effects studios.

Two key words to keep in mind when designing a production flow are **on-target** and **compromise**. Any production flow seeks to optimize the elements and process involved in a project so that the execution and completion of the goals is on-target. The challenge though is that the concept of being on-target often means very different things to each member of the production team or to each group within a larger production. For example, to the person responsible for the finances of the production, on-target usually means "within budget." To a pragmatic producer or account manager in the case of a commercial for TV, on-target is almost always equated with "the client liked it." To a perfectionist artist, on-target rarely means something other than "I am satisfied with it." It is therefore important that the individuals who are in charge of deciding how the team is supposed to deliver a project that is on-target (**by the deadline** and **within budget**) understand how to balance the different aspects of the project and also how to reach a viable compromise between the forces that drive a project. If that production balance is missing from a project, then one or several of the groups who are responsible for a specific deliverable might find the process quite grueling.

Creative Goals and Production

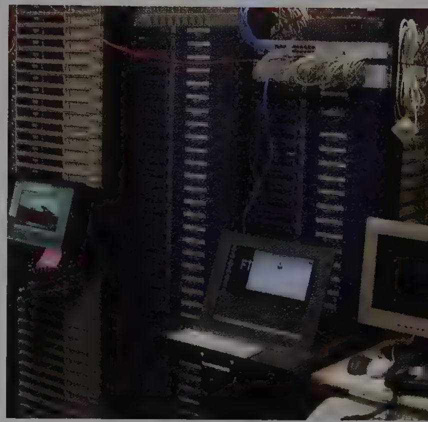
Defining the creative goals of a project is sometimes a straightforward and transparent process, other times an elusive recurring nightmare. In general, defining the **creative goals** of a project that involves three-dimensional computer animation is best done by an individual, an art director for example, who sets a **visual style** and a small group of individuals who have a common creative vision. There are no fixed rules that govern the best way to develop the creative goals of a project, but there are **best-practice guidelines**. For example, it helps to lock down the creative goals as early as possible, and it is best not to change them during production too many times or too drastically. A word of caution: Drastically changing the creative goals of a project once production has started almost always has a negative ripple effect that leads to delays, complications, additional expense, and frustration.

Budget, Schedule, and Resources

A computer animation project is defined to the greatest extent by the **resources** allocated to it. One of the main tasks, for example, of a director of visual effects or supervisor of computer animation consists of making sure that the **budget** allocated to the project by the producer and director is adequate to produce the desired results. Equally important is that the production **schedule**—also set by the producer or director of the project—is based on realistic deadlines and that it provides sufficient time to achieve the desired results. Both the budget and the schedule of a production drive much of the daily dynamics of the production because they determine the number and expertise of the **personnel** that can be hired and the

amount and power of the **computer resources** that can be used.

The amount of time that is necessary to complete a computer animation depends on the type of production and its technical complexity. The typical duration of the production of a simple commercial computer animation is between a couple of weeks and a couple of months, and such a production may employ a few animators and technical directors. A short production with complex technical requirements and compositing—for example, a television commercial—can last between two to three months and may require half a dozen animators and computers and a couple of supervisors. A longer computer animation production that involves dozens of shots and requires a lot of interaction with live action, special effects, and compositing—for example, a feature movie—may take more than one year to complete. This type of production may involve a production staff of over 100 individuals and a rendering farm with dozens or hundreds of computers. The technical challenges of experimental computer animations vary greatly. Some are produced during the course of a year by a couple of animators working after hours on a couple of borrowed computers. Other experimental computer animations—those with better funding—may be completed in six months and employ half a dozen individuals working on several computers.



2.5.3 Two views of a rendering farm with 250 dual-processor computers, 2 GB of RAM each, 4.3 TB of Fiber Channel RAID storage, and 1000- and 100-Base T network switching equipment. (Images courtesy of Render Core, Hollywood.)

2.5 The Digital Computer Animation Studio

Most computer animation projects require the collaboration of many individuals with different skills, talents, personalities, and working habits. Computer animation projects are **team efforts** where **collaboration** is a key ingredient for success. The production of any computer animation requires lots of **planning** and constant **supervision** because of the number of individuals involved, the short production cycles, limited budgets, and the unpredictable and changing nature of cutting edge technology. Improved network communication, more standard software, and better understanding of production flows have made possible the production of computer animated projects in multiple locations through **remote collaboration**.

The creation and production of computer animation takes place in environments where most of the tools and processes are computer-based. There are many ways to configure projects within a **digital studio** depending on the complexity and volume of work that needs to be done, the deadlines, the number of people working in the studio, their experience, and their talent. A digital studio needs a creative vision that focuses the activities, styles, and specialties of the group. Digital animation studios include personnel, software, and computer systems with a specific configuration of processing power, storage, networks, and input and output peripherals (Fig. 2.5.1). Last but not least, a digital studio needs to be backed up by a **business plan** that clearly defines—among other things—the balance between income and expenditures, how to deal with significant equipment and software upgrade costs, and the means by which to achieve growth.

Creative Team (TV Commercial)

Creative Director
Art Director
Copywriter
Producer
Account Executive
Animation Director

Production Team (TV Commercial)

Animation Supervisor
Senior/Junior Animators
Technical Directors:
Modeling, Rigging,
Lighting, Rendering
Roto Artist/Compositor
Producer
Production Manager
Technical Assistant

2.6.1 Sample creative and production teams in the production of a an all-computer-animation TV commercial.

(© 2002 Harald Siepermann.)

Personnel

The **personnel** of a computer animation studio or production house includes creative, technical, production, and administrative positions. The number of individuals employed is in direct proportion to the size and volume of a particular project or a studio. Small studios may employ only five individuals, medium-sized studios may have around 20 employees, and some of the large studios can have as many as 100 employees. Large studios sometimes include creative and technical personnel from areas other than computer animation, such as traditional character animation, live action film, optical compositing, sculpture, and model making. Likewise, the administrative positions in a large studio might include specialists in accounting, sales, training, human resources, and distribution.

Production studios tend to hire some people as permanent or **full-time employees** and others as **freelancers** or **contract employees**. The balance between these two types of employees varies greatly between companies and production cycles. Small studios have a tendency to hire a fair number of **generalists** who can perform a variety of tasks as needed, for example animation and compositing. Large companies tend to have more structured pipelines that require larger numbers of **specialists**, for example a Creature Assistant Technical Director, someone whose exclusive task throughout a production is to assist in the rigging and completion of character models. Figures 2.6.1–2.6.6 list some of the technical and creative positions for various kinds of projects. The quality of the creative and technical personnel in a computer animation team is usually measured in terms of their talent, experience, dedication, and productivity.

Turnkey Software

In addition to the operating system and all the utility programs associated with it, a digital studio is often centered around its **applications software** (Fig. 2.5.2). This type of software may include programs as diverse as three-dimensional computer animation, image compositing, motion capture, and digital ink and paint. A large number of computer animation facilities use **turnkey software**. This type of software, also called **off-the-shelf software**, is commercially available from a variety of vendors and is ready to use on virtually all computer platforms. Turnkey software systems can range in price from under US$1,000 to several tens of thousands of dollars depending on their capabilities, sophistication, and speed. Small turnkey systems are usually sold as a single unit, but large turnkey systems are usually sold as a collection of stand-alone modules that can be purchased in different configurations. Oftentimes the functionality of application software can be enhanced through the addition of **plug-ins** which are essentially add-on programs with a specialized set of functions. Plug-ins may range in price from a few dollars to several thousands.

When selecting turnkey software it is important to consider its upgrade policies, and its upward and downward compatibility. Turnkey software is upgraded periodically. Upgrades consist of adding new functionalities, optimizing existing features, and fixing problems—also called software bugs. In general, **software upgrades** are offered to owners of the upgraded software at a nominal fee, but on occasion the extent of the software upgrade is such that the software is considered a new version of the product. In the latter case, the upgraded software is sometimes offered to owners and new buyers at the same price. **Upward compatibility** exists when files created with previous versions of the software are compatible with new software upgrades. **Downward compatibility** exists when files created with a new software upgrade are compatible with earlier versions of the software. In addition to applications software, computer animation and visual effects houses use **operating systems**, which are programs that manage each individual computer system in the network. A few of the operating systems most widely in use today include Linux, Windows, OS X, Irix, and Solaris.

Proprietary Software

Much of the sophisticated computer animation software is often produced with a combination of commercially available software and custom, or proprietary, software. **Proprietary software** is developed in-house to provide tools and techniques that are not available in commercial turnkey systems. Proprietary software can also be used in conjunction with turnkey software; for example, it can be used to preprocess motion-capture data before it is sent to the turnkey animation module. Proprietary software is often quite costly because it requires a team of specialized and dedicated programmers to develop, maintain, and upgrade it. Using proprietary software in a project that is being split between several production houses can sometimes create complications and limit production flexibility.

Processing Power

The **processing power** of a computer animation facility is determined by the power, speed, and number of computers dedicated to compute the animation. The power and speed of a single computer system is dictated by the configuration of the computer's central processing unit, graphics processors, clock, bus, RAM memory, and peripheral storage. The exact configuration of a computer used to create animation depends both on the budget available and on the type of task assigned to that machine—for example, modeling, rendering, or compositing.

Today's production of computer animation is dominated by microcomputers with one or several processors, including the newer dual-core or quad-core desktop models. A **multicore CPU** combines several processors on a single chip. An average computer used for animation production might have between one and four 32-bit

Bunny Team	
Writer/Director	1
Producer	1
Digital Effects Supervisors	2
Lead Animators	2
Animators	12
Lighting Lead	1
Technical Directors	19
Modelers	16
Digital Paint Artists	3
Editor	1
Production Coordinator	1
Software Tools	6
Research and Development	6
Systems Support	2
Technical Assistants	2
Production Executive	1
Production Manager	1
Production Assistant	1
Production Accountant	1

2.6.2 Partial listing of screen credits and statistics from *Bunny*, an independent film directed by Chris Wedge at Blue Sky Studios that won the 1998 Academy Award in the Animated Short Films category. Notice the medium-sized team required to complete the award-winning short film.

**Creative Team
(Effects Feature Movie)**

Director
Scriptwriter
Production Designer
Visual Effects Director
Art Director
Storyboard Artist
Producer

**Administrative Team
(Effects Feature Movie)**

Executive Producer
Production Assistant
Production Manager
Director of Postproduction
Director of Finance
Production Accountant

2.6.3 Sample creative and administrative teams in a large production, a live action feature film with visual effects sequences.

2.6.4 (Facing page) Sample production team in a large production, a live action feature film with visual effects sequences.

or 64-bit CPUs, 2 gigabytes of RAM, clock speeds above 2 GHz (gigahertz), a powerful graphics card with sophisticated hardware rendering capabilities, and 1 terabyte of storage. A high-end production computer might range from four to sixteen processor cores, up to 64 GB of RAM, and 10 TB of storage. Microcomputers are also being increasingly used as the building blocks of **computer rendering farms** that include dozens or hundreds of individual processors splitting a rendering job and communicating through a fast speed network (Fig. 2.5.3). In 2001 the movie ***Final Fantasy: The Spirits Within***, for example, employed rendering farms with 960 CPUs running in the Linux environment and more than 300 CPUs in Irix.

Peripheral Storage

The type of **peripheral storage** used in a computer animation facility is also based on the volume, quality, and complexity of the work done at that facility. The frames of a computer animation are stored in digital form as they are generated and until they are recorded on film or video. Online storage is necessary so that it is possible to preview an animation in progress or to retouch and to composite some frames in the animation. The size of a single frame of high-quality computer animation may range from about 5 to 60 MB (megabytes or millions of bytes) depending on its spatial, temporal, and chromatic resolution. A 2K file, for example, destined to be recorded on 35 mm film (2048 x 1556 pixels) saved in 8-bit color uncompressed TIFF file format is close to 13 MB, almost 26 MB in 16-bit color, and over 51 MB in 32-bit color. The online storage capacity of a production facility may be measured in gigabytes (GB) or billions of bytes, and terabytes (TB) or trillions of bytes (which are too small a unit to measure online storage). The 2001 ***Final Fantasy: The Spirits Within*** movie, for example, employed about 10 terabytes of online storage for three-dimensional data and 5 TB for two-dimensional data. A standard PC computer used for production today may have a 500 GB hard disk, or access to a 5 TB desktop RAID hard disk. **RAID disk arrays** (Redundant Array of Independent Disks) are a popular way to store data safely. A RAID array is a group of drives that act as a single storage system with high levels of redundancy and fault-tolerance. There are several levels of RAID redundancy. Level 1, for example, creates a mirror image of all data using two hard disks. RAID Level 5 saves data across several drives, including the parity information that allows recovery from the failure of any single drive.

Networks

The main function of a computer **network** is to bring information to the processors from storage and the peripheral devices, and vice versa. Networks usually have one or several computers—called **network servers**—whose main purpose is to help the other computers on the network fetch and send data. **Internets** are networks that

connect computers in locations across the globe, whereas **intranets** connect computers that belong to a single company or might even be located in the same building. The **bandwidth**, or transmission capacity, of a network is a crucial issue that determines its functionality. Some popular network bandwidths include **T-1** or **DS-1** (Digital Signal Level One) at 1.544 megabits per second, Xerox's **Ethernet** at 10 megabits per second, **FDDI** (Fiber Distributed Data Interface) at 100 megabits per second, and **ATM** (Asynchronous Transfer Mode) at 154 megabits per second. Networks are commonly used in production environments to share files and to keep as many computers on the network as possible busy at all times. Very high-bandwidth networks are starting to be used for transferring computer animations between studios in different cities and even different countries.

Input and Output Equipment

The **input capabilities** of a computer animation studio are used for a variety of purposes. Flatbed scanners or digitizing cameras are used mostly for scanning images that may be used as texture maps or backgrounds during the rendering process. Film digitizers are used for digitizing entire live action sequences that are composited digitally with the computer animation. Three-dimensional scanners are used for digitizing the shape of scale models and full-size environments or the actions of human actors to be used as motion templates for an animation. The **output capabilities** of a computer animation facility include a variety of devices to record motion tests and the finished animations. High resolution electronic and laser film recorders are used to output computer animation onto film. Digital disk recorders are used to output animation onto a variety of video formats. Digital disk recorders are also a popular form of peripheral storage because they can record video in digital format and play back computer animation at standard video rates on video output devices.

2.6 Creative, Technical, and Production Teams

Most computer animation projects require one or several teams of individuals with a variety of skills and talents. Computer animation teams are put together in different ways depending on the nature and needs of the project. Often several teams are involved in the creation of a single project, especially when the project is complex and requires the participation of several companies. In some cases, a single team may handle both the creative and production responsibilities, but often these two stages of a computer animation project are implemented by separate creative and production teams. The members of both of these teams are often credited in the closing credits that may be shown at the end of an animated piece.

The **creative team** is usually represented by the individuals, design studio, communications company, or advertising agency that developed the concept and the visual treatment. This team is typically

Production Team (Effects Feature Movie)

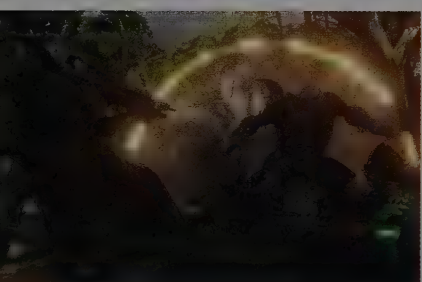
Visual Effects Group
- Visual Effects Producer
- Visual Effects Supervisor
- Visual Effects Editor
- Visual Effects Assistant Editor
- Visual Effects Coordinator
- Stage Technicians

Computer Animation Group
- Computer Animation or Digital Supervisor
- Computer Animation Shot Supervisors
- Computer Animators
- Computer Animation Production Coordinator

Modeling and Lighting Group
- Computer Modeling Supervisor
- Modeling TDs (Technical Directors)
- Lighting Supervisor
- Lighting TDs
- Rendering Wranglers

Computer Technical Support Group
- Computer Graphics (CG) Department Manager
- CG Software Developers
- CG Technical Assistants
- CG Systems Support

Digital Compositing and Postproduction Group
- Digital Supervisor
- Digital Coordinator
- Digital Artists
- Compositors
- Digital Transfer Operator
- Scanning Supervisor
- Scanning Operators
- Rotoscoping Supervisor
- Rotoscopers
- Camera Trackers
- Digital Output Supervisor
- Digital Color Timing Supervisor
- Negative Cutter



2.6.5 Screen shots and characters from the *Halo 3* game. (Halo 3™ Microsoft Corporation. Reprinted with permission from Microsoft Corporation.)

responsible for creating a script or screenplay and a storyboard. The main responsibility of the **production team** is to execute the ideas provided by the creative team and to deliver a finished product. The production team may be based in one or in several production and postproduction companies, or groups within the same company. Some of the multiple responsibilities of the **technical team** include providing technical support to the production crew, maintaining the computer systems in working order, and developing custom software for special production requirements.

Small Projects

Depending on their nature small projects may range from just a few individuals to thirty or forty people. Projects with very polished technical production values usually require significant crews. This is often the case with slick commercials for television, music videos, and even independent productions. **Crossing over** between different production roles is still common in small teams where a single individual may perform different roles and be in charge of different tasks. For example, in a small project an animator might also be in charge of modeling some of the characters and building their inverse kinematics chains, while in a larger project these three different tasks are usually performed by three different individuals. The creative and production teams for a short computer animation project, a **television commercial** for example, may include the positions listed in Figure 2.6.1. The production team may also include a visual effects supervisor and an editor if the project involves a significant amount of live action and compositing. The production team for the commercial shown in Figure 13.2.2, for example, consisted of 12 people including modelers, trackers, animators, and lighters; the schedule for the project was three months and the live action was shot in five days. Figure 2.6.2 lists most of the team members in ***Bunny***, a 1998 award-winning **independent short**.

Large Projects

The production of long computer animations may require a relatively small creative team but an extensive production team with specialized groups. Four different types of large-scale projects involving three-dimensional computer animation would include, for example, the motion-captured animation for a platform fighting game, the visual effects for a live action feature movie, an all-computer-animated feature movie, and the three-dimensional computer animation to be integrated into a hand-drawn (2D) animated feature movie.

The creative team behind the visual effects for a feature movie, for example, may consist of about half a dozen individuals including the positions listed in Figure 2.6.3. Having a large production team is one way to handle both the tight production schedules and the volume of work contained in a computer animation feature movie. A production team for this type of project may include several groups and

Halo 3

Art 35
Art Director................................1
Campaign Environment Leads.......4
Campaign Environment Artists......7
Multiplayer Environment Lead 1
Multiplayer Environment Artists....3
3D Art Lead 1
3D Artists....................................5
Animators 5
Additional Animation 1
Concept Art/Skies/Matte Painting 2
Effects Art Lead............................1
Effects Artist................................1
Technical Art Lead........................1
Technical Artists 3

Audio..3
Director/Composer 1
Audio Lead/Sound Design 1
Sound Design/Additional Music....1

Cinematics 3
Cinematics Director.......................1
Cinematics Design.........................2

Design 9
Campaign Design Leads................2
Campaign Designers 3
Gameplay Design Play 1
Gameplay Designer.......................1
Multiplayer Design Lead 1
Multiplayer Designer.....................1

Engineering 20
Engineering Leads.........................6
Campaign Engineers 5
Graphics Engineers........................4
Multiplayer Engineers....................5

Production Engineering 9
Production Engineering Lead 1
Production Engineers 2
Tools Lead Engineer 1
Tools Engineers 5

Graphic Design, User Interface.....4
Graphic Design Lead......................1
User Interface 2
Graphic Artist................................1

Production 9
Executive Producer........................1
Production Leads 2
Producers......................................4
Development Manager 1
Studio Manager 1

Test..4
Test Manager................................1
Test Leads 3

Writing 4
Writing Director 1
Managing Editor 1
Writers 2

Bungie NET Team 4
Bungie Mktng/PR/Comm. Lead....1
Web Development Lead 1
Web Developer.............................1
Oversight......................................1

Bungie Backbone.........................7
Senior Business Coordinator..........1
Business Administrator 1
IT Lead..1
Helpdesk Lead 1
Security 3

Additional Support 4
Graphic Design 1
Music Composition, Audio Prod....1
Orchestration 1
Story Editor 1

Special Thanks 21

Voice Actors 50
Cinematics Cast 10
Artificial Intelligence Cast............24
Brutes 3
Brute Chieftain 1
Civilians 2
Elites 2
Grunts 4
Marines 9
Sergeants...................................2
Multiplayer Announcer 1
Additional Voices 15
Casting/Voice-Over Prod. Svces....1

Cinematic Animation Partners....36
Animation Supervisor...................1
damnfx 26
Zoic Studios 9

Microsoft Game Studios Publishing 35
Executive Producer........................1
Legal/Business Affairs 3
Engineering...................................4
Program manager 1
Engineering Support......................7
Finance 3
HR/Recruiting 4
Research 4
User Experience 5
User Research Lead.......................1
User Research Engineers 2

Microsoft Research Asia...............6
Researchers...................................6

Microsoft Test 13
Test Lead 1
SDET...12

Microsoft Xbox..........................15
Platform and Xbox Live...............12
Marketing and PR.........................3

Microsoft..................................18
Localization Teams Europe/Brazil ..5
Localization Team Japan................4
Localization Team Korea 4
Localization Team Redmond 1
Localization Team Taiwan 4

Microsoft Development Partners 12
FASA..3
Rare 9

Producers, Engineering and Artistic Contractors.................................0
Filter 2
FilmOasis 7
Sakson & Taylor 20
Excell Data Corporation 31
Volt...63
Xversity 1

Community................................12
Biggest Community Supporters...12

2.6.6 Partial listing of screen credits and statistics of the major departments involved in the production of *Halo 3*, a 2007 game for the Xbox produced by Bungie Studios.

The Prince of Egypt

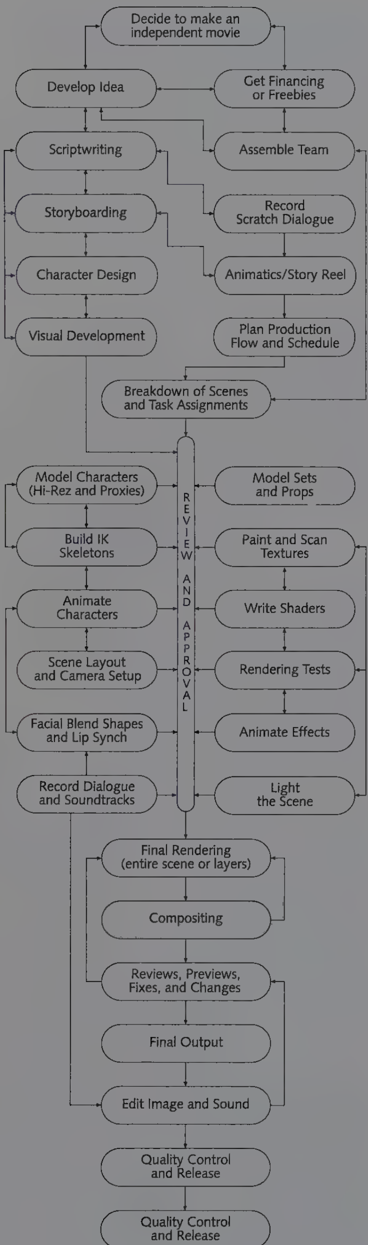
Final Line Animation (for all characters)	**111**
Department Lead	1
Sup. Character Lead	1
Character Leads	18
Senior Key Assistants	2
Key Assistants	39
Assistants	14
Breakdown/In-betweeners	36
Effects (by sequence)	**90**
2D Department Lead	1
Chariot Race, Sequence Lead	2
Animators	3
Digital Effects	4
Assistants	5
Burning Bush, Sequence Lead	2
Animators	5
Digital Effects	1
Playing with the Big Boys, Sequence Lead	1
Animators	2
Assistants	2
Players, Sequence Lead	2
Animators	2
Digital Effects	1
Assistants	9
Red Sea, Sequence Lead	2
Animators	6
Digital Effects	8
Assistants	4
Effects Breakdown/Inbetw.	19
Natural Phenom./2D CGI	2
Additional Effects Animation	4
Graphics Soft. Developers	5
Character Animation (by character)	**83**
Sup. Animator, Old Moses	1
Sup. Animator, Young Moses	1
Animators, Moses	20
Sup. Animator, Old Rameses	1
Sup. Animator, Young Rameses	1
Animators, Rameses	4
Sup. Animator, Tzipporah	1
Animators, Tzipporah	5
Sup. Animator, Jethro	1
Sup. Animator, Hotep & Huy	1
Animator, Hotep & Huy	1
Sup. Animator, Miriam	1
Animators, Miriam	2
Sup. Animator, Aaron	1
Sup. Animator, Queen	1
Animator, Queen	1
Sup. Animator, Seti	1
Animator, Seti	1
Sup. Animator, The Camel	1
Sup. Animator, Yochevet	1
Sup. Animator, Horses	1
CG Crowd Animation	7
Additional Animators	15
Additional Animating Assts.	13
Production Management (by group)	**66**
Digital Color Prod. Mgr.	1
Digital Operations Manager	1
Production Supervisors	7
Animation/Final Line	2
Effects/Story	1
Production Accountants	2
Assistant Accountants	2
Senior Coordinators	2
Editorial	1
Sweatbox	1
Coordinators	20
Story	1
Layout	2
Scene Planning	1
Animation	2
Final Line	2
Backgrounds	1
Effects/CGI	2
Scanning	2
Plotting	1
Checking	1
Color Models/Visual Dev.	1
Digital Paint	2
Film Recording	1
Script Continuity	1
Assistants	9
Production Assistants	25
Publicist	1
Editing	**42**
Digital Paint	**41**
Paint Checkers	2
Painters	39
Layout	**29**
Department Lead	1
Key Layout/Workbook	6
Layout	19
CGI Layout	2
Blue Sketch	1
Background	**20**
Background Artists	17
CGI Digital Backgd. Artists	3
Story	**16**
Writer	1
Writers	8
Add. Screenplay Material	1
Additional Story	6
Anim. Digital Final Check	**14**
Directing/Supervising	**13**
Directors	3
Executive Producer	1
Producers	2
Associate Producer	1
Art Directors	2
Production Designer	1
Sup. Editor	1
Sup. Production Manager	1
Production Manager	1
Artistic Supervisors	**13**
Story	2
Layout	1
Background	2
Visual Effects	2
Scene Planning	1
Color Models	1
Scanning	1
Anim./Digital/Final Check	2
Digital Paint	1
Character Design	**10**
Designers	3
Additional Char. Design	5
Sculpting	2
Visual Development Design	**9**
Color Models	**8**
Colorists	5
Color Markup	3
Scene Planning	**7**
Scanning	**6**
Scanners	5
Background Scanner	1

2.6.7 Partial listing (by size) of screen credits and statistics of some of the major departments involved in the production of *The Prince of Egypt*, a 1998 2D hand-drawn and 3D computer animated DreamWorks production.

Ratatouille

Animation	**95**
Animation Manager	1
Directing Animators	2
Animation Preproduction	7
Animators	63
Fix Animation Lead	1
Fix and Additional Animation	7
Crowds Animation Lead	1
Crowd Animators	4
Animation Technical Support	1
Animation Simulation Artist	1
Additional Animation	3
Animation Coordinator	1
Animation Technical Coordinator	1
Animation Fix Coordinator	1
Animation Production Assistant	1
Directing/Supervising	**47**
Director and Screenwriter	1
Co-Director	1
Producer	1
Executive Producers	2
Associate Producer	1
Original Story	3
Music Composer	1
Story Supervisor	1
Supervising Technical Director	3
Production Designer	2
Supervising Animators	2
Director of Photography Lighting	1
Director of Photography Camera	1
Character Design	4
Character Supervisor	1
Sets Art Director	1
Sets Supervisor	1
Shading Art Director	1
Shading Supervisor	1
Global Technology Supervisor	1
Effects Supervisor	1
Simulation Supervisor	1
Groom Supervisor	1
Crowds Supervisor	1
Production Manager	1
Sound Designer	1
Casting	2
Additional Story Material	3
Production Accountant	1
Lighting Supervisor	1
Matte Supervisor	1
Rendering Supervisor	1
Additional Prod. Management	1
Lighting	**47**
Lighting Managers	2
Technical Lighting Lead	1
Lighting Technology Lead	1
Master Lighting Artists	17
Shot Lighting Artists	21
Lighting Technology	3
Lighting Coordinator	1
Lighting Production Assistant	1
Sets and Layout	**39**
Set/Layout Manager	1
Senior Camera Operator	1
Layout Lead	1
Sequence Leads	3
Layout Artists	9
Previsualization	2
Layout Coordinators	2
Set Modeling Lead	1
Sets Technical Lead	1
Modeling Artists	7
Set Dressing Leads	2
Set Dressing Artists	3
Additional Sets Artists	3
Sets Coordinators	3
Production Office Support	**39**
Art	**38**
Art Manager	1
Development Art Director	1
Environment Designer	1
Additional Character Design	3
Production Artists	5
Graphic Designers	3
Sculptor	1
Matte Painters	2
Matte Technical Artists	3
Additional Visual Development	4
Graphics Translations	2
Molds and Castings (by Images in Motion)	*
Additional Production Artists	2
Art Coordinator	1
Art Production Assistants	2
Art Interns	7
Shade, Paint and Groom	**36**
Shade/Paint/Groom Manager	1
Character Shading Lead	1
Shading Artists	17
Digital Painters	6
Lead Groom Artist	1
Fur and Hair Groom	2
Additional Shading	4
Shading Packet Artists	2
Shade/Paint Coordinators	2
Production Engineering	**36**
Team Leads	5
Software Development	24
Infrastructure	7
Story	**25**
Additional Story Supervision	1
Story Manager	1
Story Artists	13
Additional Storyboarding	5
Animatic Artists	2
Script Supervisor	1
Story Coordinators	2
Crowds and Simulation	**24**
Postproduction	**24**
Music	**22**
Characters	**21**
Character Managers	2
Character Leads	4
Modeling and Articulation Artists	11
Character Scans (by Gentle Giant)	*
Add. Modeling and Articulation	3
Character Coordinator	1
Technical Development	**20**
Postproduction Sound Services (by Skywalker Sound)	**20**
Effects	**19**
Effects Manager	1
Effects Artists	17
Effects Coordinator	1
Editorial	**18**
Cast/Voice Talent	**17**
Image Mastering	**14**
End Titles	**13**
Chefs de Pixar	**10**
Rendering and Optimization	**9**
Render Pipeline Group	**8**
Additional Voices	**7**
Moving Pictures Group	**6**
Sweatbox	**2**

2.6.8 Partial listing (sorted by size) of screen credits and statistics of some of the major departments involved in the production of *Ratatouille*, a computer animated Pixar Animation Studios production and winner of a 2007 Academy Award.



2.7.1 A possible production pipeline for a small team working on an independent computer animated short. The vertical bar in the center of the flowchart represents the review and approval process.

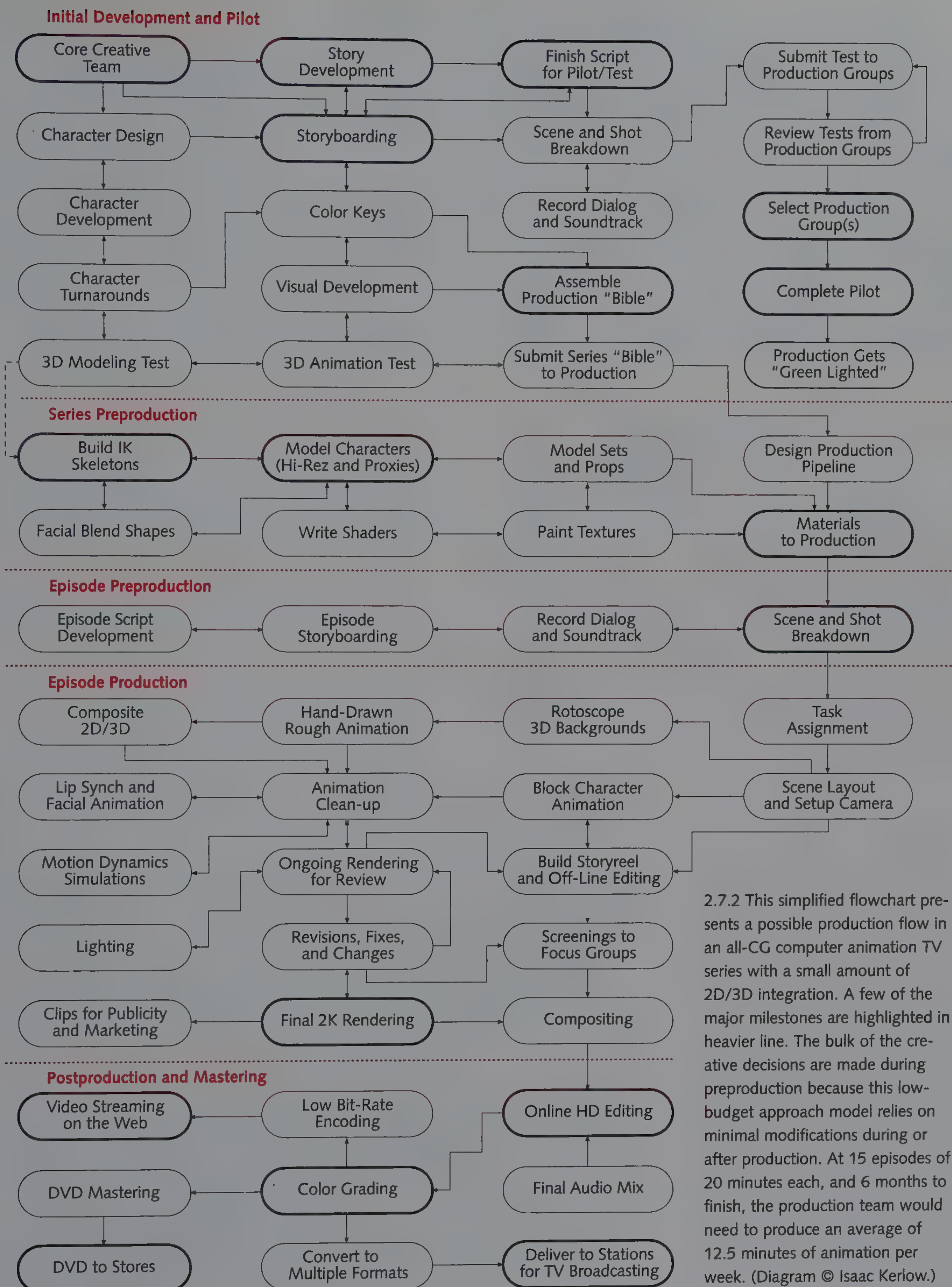
positions (Fig. 2.6.4), but under some circumstances a few responsibilities may be assigned to a single individual. Crossing over between production roles, however, is still rare in large projects, where individuals tend to be more locked into their specialties.

The **visual effects** group is responsible for the overall production of all the special effects in the project, including the supervision of the computer animation. Usually the director decides on the creative treatment for every single shot that requires special effects. A visual effects producer develops production guidelines, as well as a budget and a schedule for the project. A visual effects supervisor specifies the **production techniques** to be used and is also in charge, along with the visual effects producer, of making sure that the production guidelines, budgets, and deadlines are followed by the different production groups and subgroups. He or she functions as the liaison between the live set and the digital facility. (See Chapter 13 for additional information on the visual effects production process.)

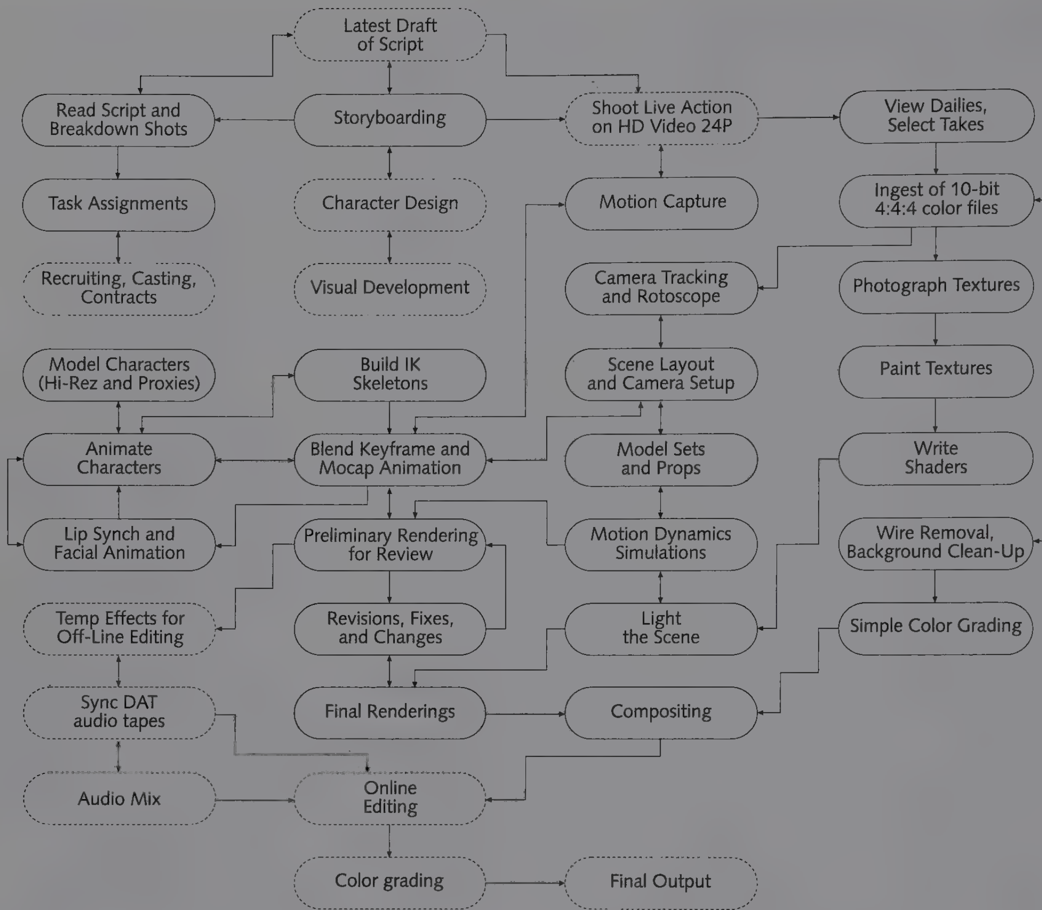
The **computer animation** group is responsible for the production of the computer animation sequences. In the case of a visual effects movie, the computer animation supervisor primarily makes sure that all the visual effects guidelines are understood and implemented by the animator. In the case of a computer animated movie, the animation director sets the tone and the target for the performances that animators have to extract from their characters. The computer animation shot supervisors are in charge of subgroups that are responsible for completing a single shot or a series of shots. The computer animators develop the imagery and motion tests until the sequences are approved by the visual effects director. The computer animation production coordinator makes sure that everybody has what they need to do their job; this individual also schedules equipment, personnel, and meetings. **Shot assignments** and footage quotas are usually made by a group of people that may include the animation director, producer, and project manager. Animators are assigned shots based on skill, sensibility, workload, or continuity. Some animators are better at action scenes or physical comedy, while others might be better at emotional delivery and subtle introspective moments. Animators are sometimes assigned to a single character throughout the movie (Fig. 2.6.7) or be allowed to animate all the characters in the shot (Fig. 2.6.8). **Footage quotas**, also known as frame or animation quotas, are the number of frames (there are 16 frames in one foot of 35 mm film) that an animator is expected to deliver on a weekly basis. Footage quotas are usually related to the experience of the animator, the quality of the animation tools, the complexity of the scene, and the schedule. In *A Bug's Life*, for example, the quota averaged 2.5 feet a week (40 frames) per animator, while in *Monsters, Inc.* it averaged 4.5 feet (72 frames) a week.

Last but not least, the **digital compositing and postproduction** group may be responsible for scanning backgrounds, retouching, tracking cameras, rotoscoping, compositing, and outputting the different layers of visual effects, computer animation, and live action.

In addition to the creative, technical, and production teams, the

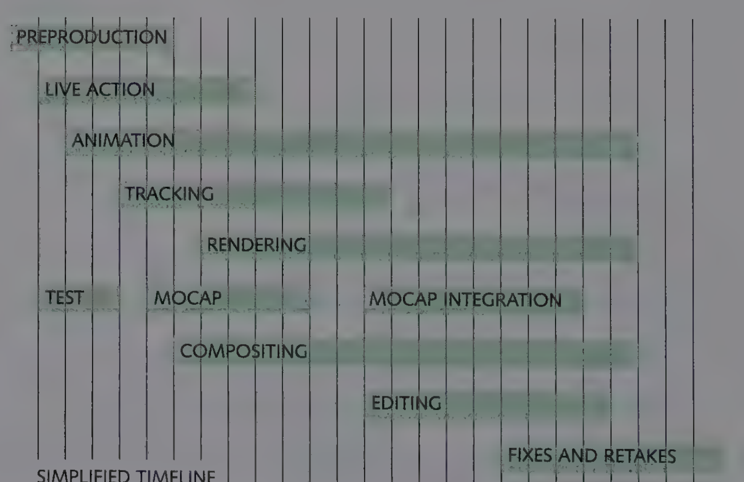


2.7.2 This simplified flowchart presents a possible production flow in an all-CG computer animation TV series with a small amount of 2D/3D integration. A few of the major milestones are highlighted in heavier line. The bulk of the creative decisions are made during preproduction because this low-budget approach model relies on minimal modifications during or after production. At 15 episodes of 20 minutes each, and 6 months to finish, the production team would need to produce an average of 12.5 minutes of animation per week. (Diagram © Isaac Kerlow.)



2.7.3 (Above) This flowchart presents a simplified production flow in a visual effects movie shot on HD with 3D animated characters, motion capture, and camera tracking of the live action HD background plates. Notice how many of the major tasks are interrelated, and most of them run in parallel. Tasks inside the dotted line globes are performed by individuals outside of the core computer animation group. (Diagram © Isaac Kerlow.)

2.7.4 (Right) The horizontal color lines in the simplified project timeline represent examples of parallel tasks tracked on a weekly basis.



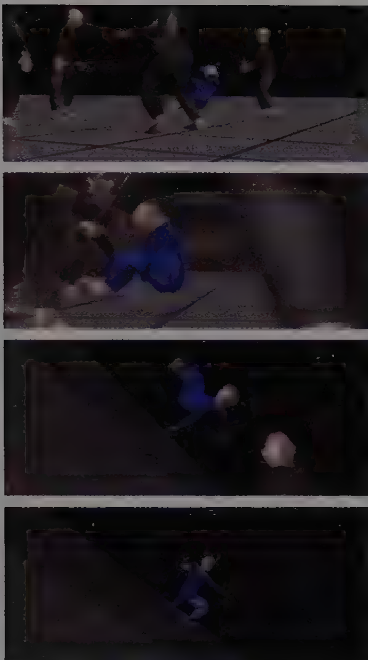
administrative team oversees many of the financial, legal, and marketing issues related to the production of a complex computer animation project. An administrative team typically involves the positions listed in Figure 2.6.3. The members of the administrative team also work with the group directors in the creative and production teams to make sure that production budgets, deadlines, and strategies are adequate to complete the project successfully. Figure 2.6.7 also lists some of the team members in the production of a **hand-drawn animated feature movie** with significant three-dimensional computer animation. Figure 2.6.8 lists some of the team members in a **computer animated feature movie**. Notice the different approach to having animators animate characters: in one case animators are assigned specific characters for the entire production while in the other one animators may work on whichever characters appear in the scenes they are assigned to animate. Figure 2.6.6 lists the team members in a large gaming project, including the group in charge of creating the **cinematics**. Notice the size of the programming team in relation to the art and animation teams (Fig. 2.6.5). Figure 13.1.3 lists the majority of the crew that is necessary to produce a major visual effects movie. The listing includes the main supervisory roles as well as the separate groups in charge of live action and visual effects production.

2.7.5 Storyboard drawings that emphasize facial expression, character pose, intention, and scene layout. See Figure 2.2.6 for the rendered version of the *Monsters, Inc.* drawing (top) and Figure 12.4.1 for the *Shrek* drawing (bottom). (Top: *Monsters, Inc.* © Disney Enterprises, Inc./Pixar Animation Studios. Bottom: *Shrek*™ and © 2001 DreamWorks L.L.C.)

2.7 The Production Process of Computer Animation

There are multiple styles and approaches for producing computer animation or digital visual effects, and this process is in constant transformation based on factors like creative goals, budget, schedule, and resources. In practice this process or **production pipeline** adopts different forms and variations, but the basic stages include preproduction, production, and postproduction. In Figures 2.7.1–2.7.3 you can notice a few obvious differences between the production flow in a computer animation movie, visual effects-oriented live action movie, and a platform game. A key concept to keep in mind regarding the production process is that while it is structured it is not strictly linear. This means that there is a series of steps that must happen in a certain sequence, but some of these steps happen in parallel. The main moments in the creation of computer animation include preproduction, modeling, rigging, layout, animation, rendering, compositing, and a variety of postprocessing techniques as required.

Preproduction involves all the conceptualization and planning that takes place before a computer animation or effects project is



PREVISUALIZATION

LIVE ACTION

produced. This stage in the process includes nonvisual tasks such as screenwriting, casting, and planning the management of the project, as well as visual tasks such as storyboarding and developing the overall visual look of the project. A project involving live action would also require meetings with the cinematographer and scouting for locations. Preproduction is the foundation of a project. Inadequate or erroneous preproduction usually results in delays, cost overruns, and creative deficit.

In simple terms the **production** stage in the process of three-dimensional computer animation involves a series of standard steps: modeling, rigging, animation, and rendering. First, the characters, objects, and environments used in three-dimensional computer animations are modeled. This computer **modeling** can occur with a wide range of computer-based three-dimensional techniques such as using virtual modeling tools to sculpt objects, or using a three-dimensional digitizer to capture the shape of a physical model directly into the computer program. Once the virtual actors and objects are modeled, they can be attached to the motion skeletons that will be used to animate them with a wide variety of techniques—this is called **rigging**. Modeling is covered in Chapters 3–5, and rigging is covered in Chapters 5, 11, and 12. Computer **animation** techniques range from keyframing animation where start and end positions are specified for all objects in a sequence, to motion capture where all positions are fed to the objects directly from live actors whose motions are being digitized. The results of the animation can be previewed in the form of digital flipbooks displayed on the screen. Once the objects are modeled and animated, they can be rendered. To minimize time and budget complications, **motion tests** are often produced to preview the computer animation sequences before the final production takes place. Computer animation principles and techniques, including visual effects, are covered in Chapters 10–13. Computer **rendering** is the process of representing visually the animated models with the aid of a simulated camera. The lighting of the scene and the shading characteristics are often specified before the animation is laid out, but the rendering itself, the calculation of the finished computer animated images, always happens after the modeling and animation parameters have been defined. Rendering is covered in Chapters 6–9.

Once the images have been rendered, a variety of **postprocessing** and **postproduction** techniques can be applied to the images before they are recorded. For example, computer-generated images can be digitally composited or mixed with other computer-generated images or with live action. Computer animation can also be distorted, retouched, processed, or color corrected using postproduction techniques. When computer animations are completed, they are usually recorded on videotape or film so that they can be shown later on a TV screen or in a movie theater. Each film and video format has unique requirements and characteristics. For example, the standard rate of display of animated images recorded on videotape is 30

frames per second; on film it is 24 frames per second. In many instances computer animation is delivered in a **digital format** that may be played back in real time as part of a PC, game platform, or online interactive game. Postprocessing and final output are covered in Chapters 14 and 15.

Following is an overview of the major stages in digital production of three-dimensional computer animation and computer-generated visual effects. Keep in mind that there are many ways to implement each of these tasks and that the order of execution may vary depending on a variety of factors. The production process is never entirely linear, and many tasks are interdependent, overlap, and take place in parallel. A few of the main differences between a computer animation flow and a visual effects flow are highlighted in Figures 2.7.1–2.7.3.

2.7.7 The previsualization for the movie *Iron Man* helped the creative team to establish camera positions and moves, as well as to stage the action in the shot. Notice the similarity between the previz (above, bottom frame) and the final shot (above, top frame). (Images courtesy of Pixel Liberation Front. © 2008 MVLFFLLC™ and © 2008 Marvel Entertainment. All rights reserved.)

Preproduction Tasks

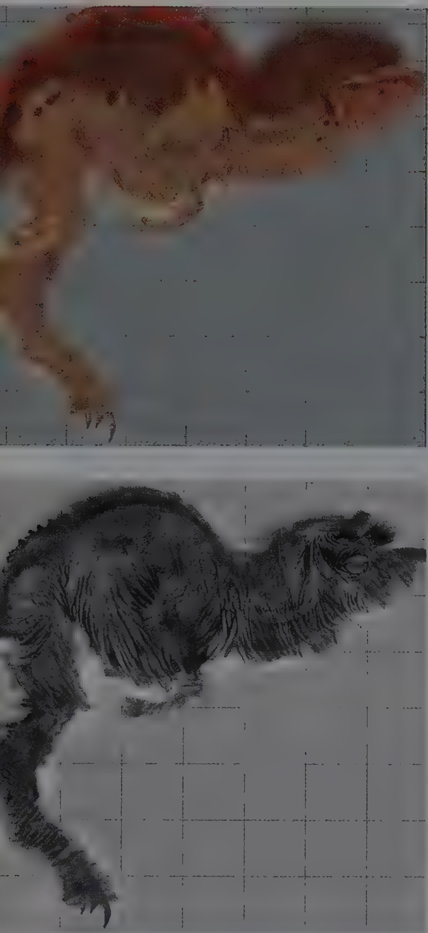
Getting ready for production varies from project to project and it generally includes visual tasks like creating storyboards and animatics, and nonvisual tasks like recording dialog. Once the script is finished, a **production-ready** version is issued to the team. This version might keep changing throughout production but it is used during preproduction for drafting the order in which major sequences and scenes will be produced. The production-ready script is also used for recording the early version dialog, called **scratch dialog**, usually recorded by non-professionals or by voice actors who might not be the ones recording the final dialog.

Before production starts the script is analyzed and "broken down" into sequences of shots and a draft production schedule. In the case of a character animation project, actors have already been cast or are cast at this point so that the final **dialog** can be recorded since it drives so much of the animation work.

Storyboards

The creation of early storyboards helps to translate the story and the script into images. In that sense, creating the storyboards is an important tool for developing the story, and also for breaking down the script into manageable production units and doing a technical breakdown of each shot. During preproduction, storyboards are partly used to fine-tune the storytelling and the timing of the action (Fig. 2.7.5). As this process evolves the storyboards generally include more production-oriented details such as the final staging, composition, actions, and camera moves. (Read Chapter 10 for more information on the different types of storyboards.)

2.7.6 (Previous page) Previsualization for a fighting sequence in the movie *Shinobi*, and the corresponding frames of composited live action with visual effects. (© 2005 SHINOBI Film Partners. Images courtesy of Links DigiWorks Inc.)



2.7.8 An image map on the top (1,200 × 1,095 pixels) and a bump map on the bottom to be applied to the model in Figure 2.7.9. (© 1999 Oddworld Inhabitants, Inc. All rights reserved.)

(Facing page bottom: *Monster Samurai*. © Sprite Entertainment, Inc.)

Animatics, Story Reels, and Previsualization

Storyboards are often used to put together an animatic, which is a collection of simple moving images used to visualize how the final project may be structured and timed. A **two-dimensional animatic** can be created by scanning single drawings from the storyboard and creating a sequence of images in time. This type of animatic often includes simple camera moves (like pans and zooms) to enhance the narrative and flow of action. A **three-dimensional animatic** goes a step further by using preliminary visual materials like wireframe or low resolution motion tests to visualize a rough cut of a computer animation. None of the special effects in an animatic are meant to be final, and often they are implemented with techniques that are cruder and less expensive than the techniques that are planned for the finished project (Fig. 2.7.6). Simple hand-drawn sketches and still photographs are common replacements for complex dynamic effects when these are presented in the form of an animatic.

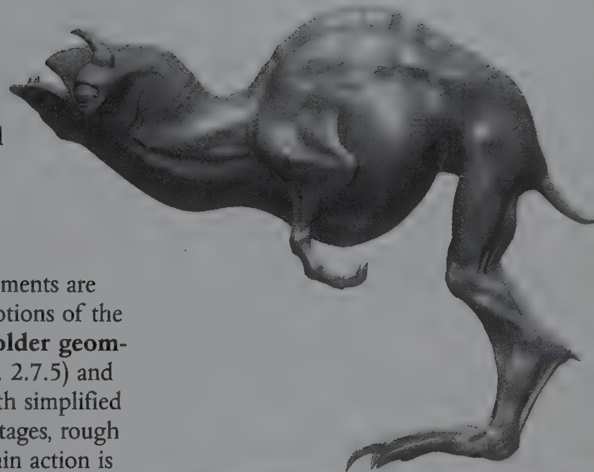
Animatics are matched to the scratch dialog and other recorded or synthesized sounds such as music and sound effects. Animatics are like motion sketches, and it is because of this that motions of animated characters in an animatic may be jerky and only insinuated. Animatics are also called **story reels,** and they are commonly shown to a client or an executive producer for review before production and as production starts. In today's production environment, animatics are usually assembled from digital flipbooks and simple live-action sequences that are composited digitally. It is partly for that reason that animatics are increasingly produced in-house at the same place where computer animation is created; both exist in a digital format. Animatics, or story reels, were known as **Leica reels** in the days of hand-drawn animation, because they were recorded with Leica equipment. The Leica reels used to evolve with the production and were regularly updated by an editor who replaced early versions of the storyboard artwork with the layout artwork, with successive versions, and eventually with the final version of the project.

In live action projects involving visual effects a **previsualization,** or **previz test**, is an approximation that is made of the models and action as close as possible to the final motion (Fig. 2.7.7). Previz is crucial in planning sequences in a movie, especially those that may be particularly complex to produce on a technical level.

Modeling

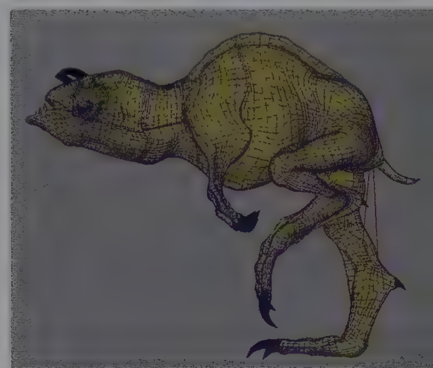
The task of modeling the geometry in a computer animation project is usually divided by type of model or by scene. In the former approach, popular with large projects, the models of the primary and secondary characters are assigned to a set of individuals or team while the props and the environments, called set dressing, is assigned to others. In the latter approach, which is perhaps better suited for simpler projects, the building of models for a particular shot is the

responsibility of the individual or team in charge of that shot (Fig. 2.7.9). It is common to build **proxies**, or placeholder geometry, early in the process so that animators, riggers, and others can get started while the final models are completed.



Rough and Final Scene Layout

While final models for the characters, props, and environments are being built (sometimes even before) the positions and motions of the characters and the camera are blocked out using **placeholder geometry** (Fig. 2.7.10). This stage is called **scene layout** (Fig. 2.7.5) and it is increasingly being done directly in the computer with simplified proxy digital models. Layout is commonly done in two stages, rough layout and finished layout. During **rough layout** the main action is blocked out with the intention to convert a story reel into three-dimensional animatics. This **animation blocking** may include the torso, simple arm movements, and head turns; anything that has to do with body language, pose, posture, and gesture. Animation is usually done from inside out: first the torso, then the arms, hands, fingers, and facial animation is last. Once staging, timings, and camera POVs indicated in rough layout are approved, the **final layout** is put together with the final files and scenes to be used for the production of the shot. Some productions still make use of the traditional approach to scene layout, which is based on drawings that detail the composition of the shot within a specific animation field (Fig. 7.2.2). The motion of the characters and the camera is commonly indicated in these drawings with directional arrows (Fig. 3.4.9).



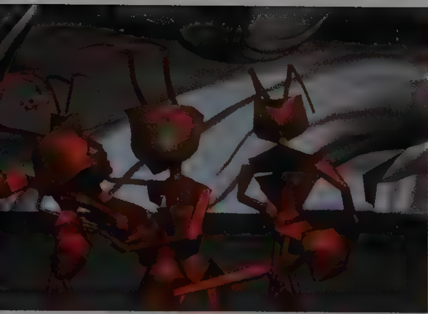
2.7.9 Wireframe and simple shading versions of a walking keyframe. (© 1999 Oddworld Inhabitants, Inc. All rights reserved.)

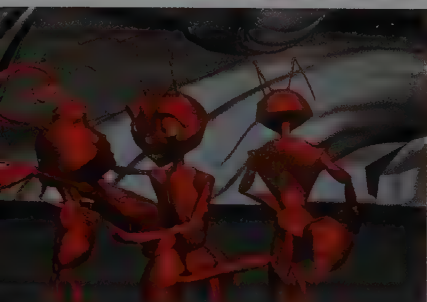
Rigging

The internal skeletons that are most often used to animate characters are called **animation rigs**. These controls are also called **IK rigs** because many skeletons are rigs or chains of controls that use the inverse kinematics (IK) animation technique (Figs. 5.7.3, 5.7.8, and 5.7.9). In projects with a large volume of animations the setting up of these animation controls is usually the responsibility of a **technical director** while the animators devote themselves to animating after the basic controls are set up (Fig. 12.5.5).

Texture Painting

Most three-dimensional computer animation projects make extensive use of textures for the geometry. Many of these textures are painted by hand directly with a digital paint system or with traditional materials and then scanned (Fig. 2.7.8). Photographic textures that are captured on film need to be scanned, and those captured with digital cameras are just transferred digitally to the computer system. Photographic textures usually require some touch-up or color bal-





2.7.10 Proxy models are used to sketch the primary animation (top), and more detailed models are used to animate the facial expressions (bottom). (Images from the motion picture *ANTZ*™. © DreamWorks L.L.C. and PDI, reprinted with permission of DreamWorks Animation.)

ance before they can be applied to the geometry. **Texture painting** and touch-up is usually done by painters, often shared with the visual development team. In some instances the texture painters have to work with the technical directors who are in charge of developing procedural textures (Figs. 9.6.4–9.6.7).

Character Animation

Three-dimensional character animation done with keyframing techniques starts with rough animation, which consists of blocking the broad motions using placeholder geometry (Fig. 2.7.10). The secondary actions and the facial animation are added next, and finally the details and the timing of overlapping motion are fine-tuned. In some studios animation details and technical animation (like hair and clothing) that are done usually after character animation is complete or almost complete are called **animation finaling**. The character animation tasks are usually divided among different individuals or teams. A common way to parcel out the character animation deliverables is by character or by scene. For example, an animator gets to animate the same character throughout many scenes, or an animator animates all the characters present in a single scene. The animation of characters in crowds is usually assigned to a separate technical animation team. **Flipbooks** contain many still frames and are useful to preview and fine-tune the animated sequences (Fig. 2.7.12).

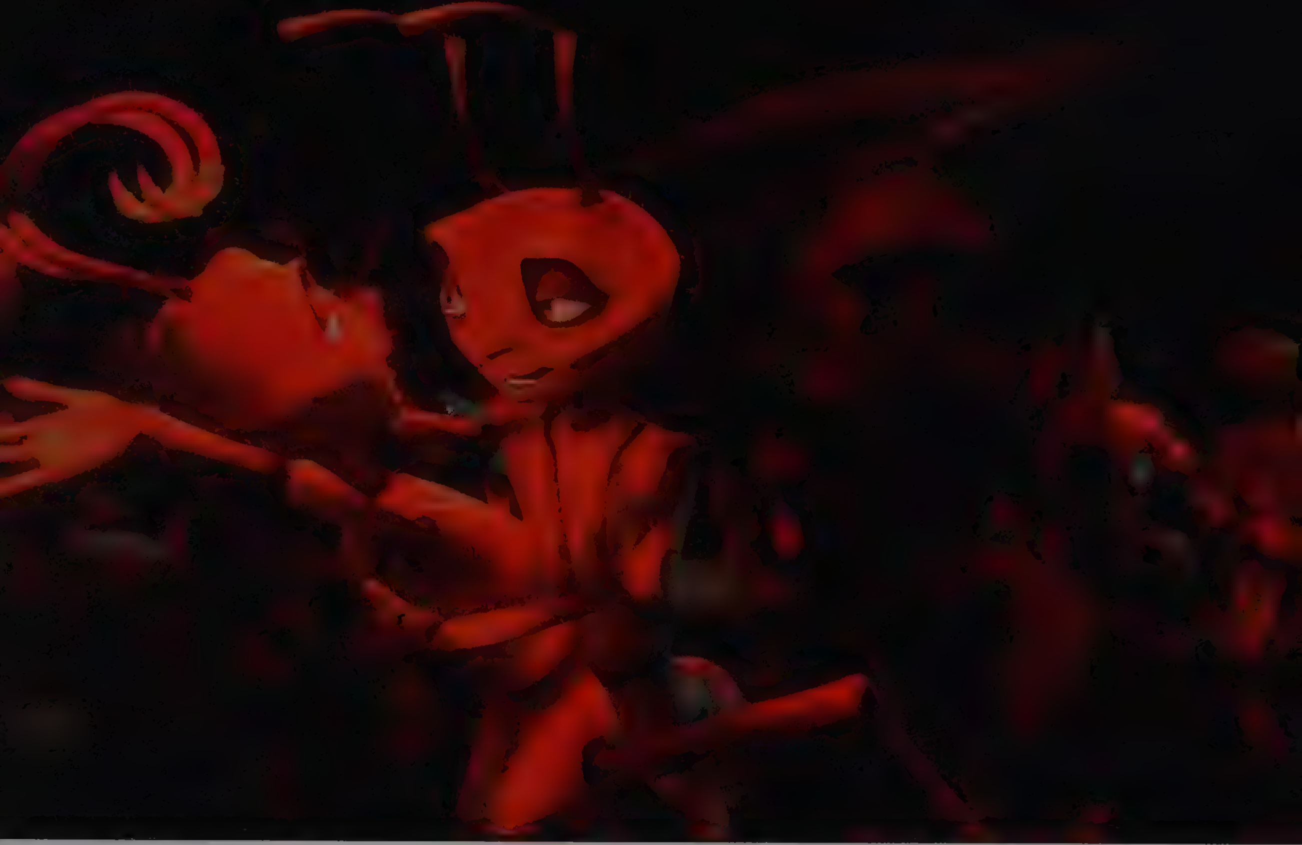
Effects Animation and Technical Animation

The **effects animation** team is typically in charge of animating the natural phenomena, like rain, wind, and fire. Most of the techniques used to animate these types of effects are procedural and require some or much scripting, therefore this task is also referred to as **technical animation**. This team is usually composed of technically oriented animators, and in some productions they also take care of animating crowds, clothing, and hair (Figs. 2.7.11 and 12.3.4).

Review and Approval of Dailies

Animation **dailies**, also called **rushes**, refers to the frames or footage animated throughout the day, or some other length of time. The term comes from live action filmmaking where footage shot daily is rushed to the lab, developed and synched to sound at the end of every day so that it can be viewed the next morning. In an animation production dailies are reviewed by the director regularly, often daily, and his/her comments and directions are transcribed as **director's notes** and distributed to the relevant members of the team. Dailies may include rough animation or renderings. Dailies are sometimes reviewed in a screening room with all or most of the animation team, sometimes in the cubicle of individual animators along with animation leads and supervisors. During dailies' review some

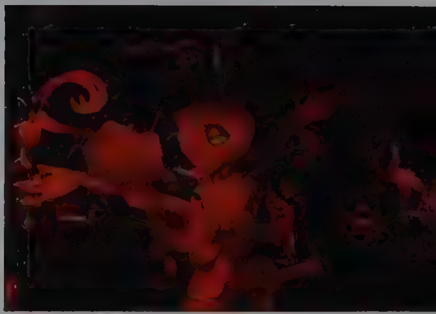
(© 2008 TeamTO–France 3–Cake Entertainment.)



work is returned to artists, animators, and technical directors for further refinement and some is considered **approved animation**.

Performance and Live Action Capture on the Set

Some character animation is partly or completely driven by the performance of live actors. These performances are captured with a variety of motion capture techniques requiring specialized equipment and software, and are covered in Chapter 12 (Figs. 2.7.13, 2.7.15, 12.1.10, 12.2.3, 12.2.7 and 12.2.8). A cinematography crew is in charge of recording the live action for a movie with film or high-definition video cameras, and some of that footage is used in the creation of visual effects (Figs. 13.1.1 and 13.1.2). Live action is sometimes also recorded as reference footage for keyframe animation, or to capture specific elements that will aid in the production of visual effects. Actors, props, or entire three-dimensional terrains are sometimes scanned for modeling and animation purposes (Fig. 5.6.2).



2.7.11 The lighting of the scene is finalized and rendered after most or all of the animation is in place. The frame on top includes motion blur and depth of field rendering refinements. (Images from the motion picture *ANTZ*™. © DreamWorks L.L.C. and PDI, reprinted with permission of DreamWorks Animation.)

Lighting and Rendering

Lighting and rendering are sometimes done by different teams. **Lighting** involves the placement and fine-tuning of all the light

2.7.12 Two versions of an animated 12-frame walk cycle. The right leg in the top version stays in the air a little longer than the version in the bottom. (© 1999 Oddworld Inhabitants, Inc. All rights reserved.)

2.7.13 The motion of this athlete is captured and will later be applied to a three-dimensional animated figure. (Courtesy of Angel Studios.)

sources in every shot of the film. The nature of lighting in computer animation is very close to traditional cinematography. The lighting stage in many productions is the first time that the detailed geometry is all placed in the scene. Lighting is often fine-tuned or touched up during the compositing stage. In fact it is common for studios to employ the same individuals as both lighters and compositors. **Rendering** involves the shading of the geometry and oftentimes the developing of special shaders that are used for this purpose (Fig. 2.7.11). As with the creation of procedural textures, some aspects of lighting and shading are often done by technically inclined artists or by technical directors who are able to program. The rendering process is often a direct result of the look development stage when most components that impact the final look of the rendered images are defined. Jobs are sent for final rendering, often with the aid of queuing software, when all the rendering tests have been approved and when all the rendering variables are set (Fig. 2.7.14). Small files are often rendered in one pass, and all the elements are rendered at the same time. Larger files with complex geometry or complex renderings are often rendered in layers or multiple passes that require further compositing, each with different elements of the shot.

Rotoscoping and Camera Tracking

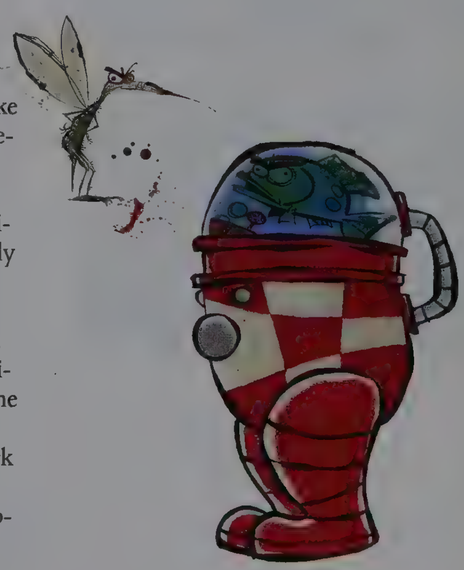
Visual effects projects consistently require the matching of computer-generated elements to live action elements, and vice versa. Much of this matching is usually accomplished through rotoscoping and camera tracking techniques (Figs. 13.2.1, 13.2.2 and 13.3.1). The position and motion data extracted from live action still frames is used by the animators and compositing artists as essential reference for their work. Rotoscoped live action, for example, can be matched to two-dimensional or three-dimensional animation.Both rotoscoping and camera tracking can be done manually or in a semi-automated mode. (See Chapter 13 for more details on these techniques.)

Compositing, Postprocessing, and Final Output

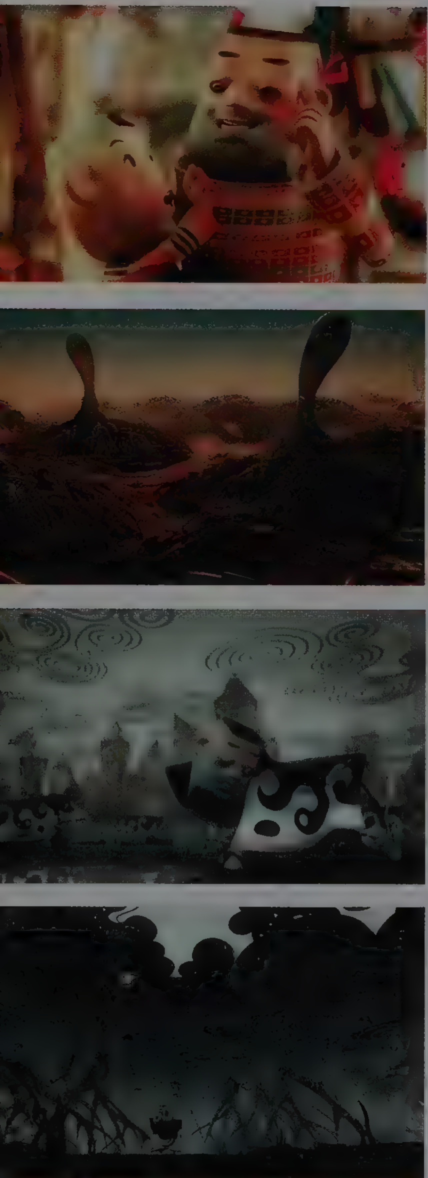
After the elements of a shot or a scene have been rendered, they are assembled and composited before the final recording takes place (Figs. 14.2.7 and 14.5.1). Stills or sequences that might need color balancing, retouching, or correcting are also taken care of during this stage (compositing and retouching techniques are described in Chapter 14). The logical culmination of the computer animation production process is the recording of the image onto a wide variety of media including film, videotape, DVDs, and paper (the basic concepts of output are covered in Chapter 15).

Media Asset Management and Technical Support

An often invisible but crucial component of the digital production flow is the management of the different art or media assets that make up a computer animation production. While there are several off-the-shelf **media asset management** software programs, some of the main challenges remain the constant revision of assets and the large number of people often involved in generating a single frame of animated action or visual effects. Media asset management requires daily discipline; an easy-to-understand system for identifying files from different scenes, shots, and revisions; frequent **digital backups** of the data; and the ability to share files between multiple online users. The technical support group is a key component in all computer animation productions. These individuals are not directly involved in the creation of the art assets—or, in Hollywood lingo, they do not "touch the product." But without their diligence, vigilance, and quick response to technical problems, most computer animation projects would be truly unmanageable. The core mission of the technical support group is to make technology invisible, and to make sure that everything works so that the creative team can do their job and the production can be completed on time.



(© 2002 Harald Siepermann.)



2.7.14 Final rendered frames from the *Kudan* animated short. (Director: Taku Kimura. Animator: Koichi Yamagishi. © 2008 Links DigiWorks Inc.)

2.8 Getting Started

Being a successful computer artist, animator, or technical director requires a lot of different skills and talents. Some are in the creative front, a few in the technical area, and many in a variety of other areas. The specific balance of skills required varies among the industries that use three-dimensional digital technology for art production: feature and television animation, platform and location-based interactive games, and websites.

Digital technology has been changing the face of art and animation production in a significant way since the late 1980s. The day-to-day aspects of animation production seem to be in a slow but constant state of flux as changes continue to take place in production, creative, and business areas. Nevertheless there are some basic strengths that artists, animators, and technical directors in this field aspire to have, or that employers look for in potential employees. The ten tips listed in Figure 2.8.2 have been distilled from years of experience and conversations with both novice and experienced practitioners. These tips might help you achieve your professional goals faster, make your career path smoother and more successful, and keep an eye on health issues (Fig. 2.8.1) such as potential **repetitive stress injury** that can result from working with computers for very extended periods of time. Download the complete *Ten Career Tips* text from **www.artof3d.com**.

Assembling a Demo Reel

As mentioned earlier, a demo reel is a compilation of your best animation and effects work, and it shows a prospective employer, client, or colleague what you are capable of both technically and creatively (Fig. 2.8.4). The first step in putting together a demo reel is deciding what content and techniques you want to showcase (Fig. 2.8.3). This, of course, depends on what type of skills you have and what type of positions you are applying for. Your demo reel should include a lot of character animation if, for example, you are interested in a character animation position. A demo reel with examples of compositing and technical rendering is unlikely to get you a job as a computer animator.

The second step consists of finding out what the employers' application requirements are and what job openings or skills they might be looking for when you apply. You can easily do this research on the Web or by phone, and from it you should get useful information about the type and length of work to include.

The third step consists of reviewing the material that you would consider including in the demo reel, and selecting the segments that work best and represent you the best. Show it to others, especially individuals not familiar with your work, and get their opinions and reactions. After the initial selection you have to decide what would be the most effective sequencing to showcase the work. Editing a demo reel is a process that requires some expertise with video editing pro-

grams. Fortunately this can be done digitally with programs such as Apple Final Cut Studio, Adobe Premiere, Avid DV Express, or Media 100. The selection and editing process might take several days and it usually requires a couple of different cuts before you find the one that looks good. Selecting a music soundtrack that works with your sequence of images and creating titles at the beginning and end of the reel are also part of putting together the demo reel. These titles should be easy to read and should include your name, telephone number, and email address.

The fourth step in putting together a demo reel is to record and duplicate the edited materials on a particular media. You can present your demo reel on a videotape, DVD, CD, or a website. If you go for the traditional videotape keep in mind that VHS and DV are the most popular video formats for demo reels because practically everyone in the computer animation and visual effects industries has easy access to a VHS and/or a DV player. Other video formats, including HDV, DVCAM, Betacam SP or SX, and Digital Betacam, offer better quality than VHS or DV but not everybody has an easy way to play them: double-check before sending them. Also keep in mind that different countries use different video standards. In a nutshell the NTSC standard is used in the United States, Japan, and most of Latin America. The PAL standard is used in most countries in Europe and much of the world. SECAM is used in France and a few other countries. Multi-standard players are common in Asia and Europe.

2.7.15 Davy Jones and his cursed crew from *Pirates of the Caribbean* were designed as characters with multiple incarnations: humans, skeletons, and marine creatures. (© Disney Enterprises, Inc. and Jerry Bruckheimer, Inc. All rights reserved. Computer animation by Industrial Light & Magic.)

2.8.1 The ideal posture to avoid repetitive stress injuries, backaches, or eye strain includes keeping your wrists straight, shoulders relaxed, monitor at eye-level, and frequent stretching.

Ten Career Tips

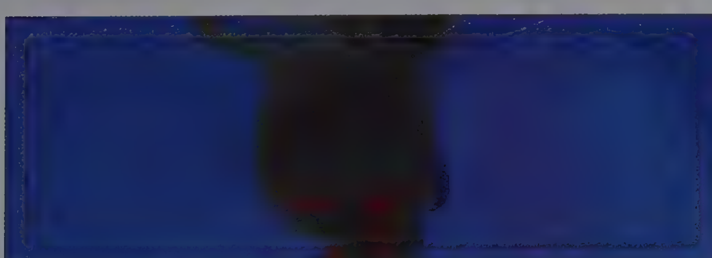
1. Be prepared for change
2. Focus on a realistic goal
3. Know your digital craft
4. Update and customize your reel and portfolio
5. Be prepared to work as a member of a team
6. Develop an appreciation for preproduction
7. Focus on issues that may impact your health
8. Learn about the history of digital creation
9. Learn about the business aspects of your career
10. Continue to develop your artistic vision

2.8.2 Ten career tips for computer animators and digital artists.

(See Chapter 15 for more information on video output.)

If you choose to present your work on a CD or a DVD, do your best so that the versions you burn can be played on different players, ideally a consumer DVD player, a PC, and an Apple computer. If your digital media will only play on a specific type of player or computer, make sure to indicate those restrictions on the label or the box. If you choose to display your work on a website you will probably need to go through an additional step of compression. This will help to optimize the material for playback and/or to fit all the material in a particular media or bandwidth. Always try to create materials that will easily play on as many players and browsers as possible. Quicktime, for example, is a popular and relatively cross-platform file format for delivering demo reels on CDs and websites.

In addition to the visuals it is useful to include some written materials that complement your demo reel. These include cover letter, resume, and the technical notes. The **cover letter** or introductory email helps employers understand what type of a job you are looking for and what special skills or needs you might have. The most effective cover letters are short and to the point. The **resume**, or curriculum vitae, provides information about your professional experience, education, and computer proficiency among other things. Listing what specific tasks you were responsible for in a specific job or project is important. Listing software programs that are part of your creative toolbox is useful, especially when you also specify your level of expertise. For example: "I am an expert user of Maya 9, quite good at Adobe Photoshop CS4, and have used a Discreet Inferno system a few times." **Technical notes** are often quite effective in providing information that may enhance the viewers' understanding of your work. These notes may include technical or practical information, for example, about circumstances or techniques that defined, constrained, or propelled your projects. Written details about the modeling, lighting, rendering, or compositing process, for example, can shed a lot of light on your skills and understanding of the technique.



2.8.3 Concept development artwork for *Oktapodi*, a student animated short nominated for an Academy Award. (Images courtesy of Gobelins and Thierry Marchand. © 2007 Oktapodi.)

Demo Reel Checklist

- ❑ Select only your best work
- ❑ Include work that shows your creativity, experience, and technical skills
- ❑ Include work that is relevant to the position(s) that you are applying for
- ❑ Highlight the work that might be of most interest to the interviewers
- ❑ Minimize editing that may distract from your work
- ❑ Include your name and contact information
- ❑ View the copies before you send them to make sure the are of good quality
- ❑ Include technical notes that help understand the work

2.8.4 Simple checklist to prepare a demo reel.

Chapter 2

Key Terms

Administrative team
Animatics
Animation
Animation blocking
Animation finaling
Animation rigs
Applications software
Approval
Approved animation
Artistic vision
ATM (asynchronous Transfer Mode)
Bandwidth
Best-practice guidelines
Budget
Business plan
By the deadline
Cinematics
Classical story design
Climax
Collaboration
Color keys
Computer animation
Computer animated feature movie
Computer rendering farms
Computer resources
Concept art
Contract employees
Cover letter
Creative goals
Creative team
Creative vision
Crossing over
Dailies
Deliverables
Delivery media
Demo reel
Dialog
Digital backups
Digital format
Digital compositing and postproduction
Digital studio
Director's notes
Downward compatibility
DS-1
Effects animation
Ethernet
Exposition
FDDI (Fiber Distributed Data Interface)
Final layout
Flipbooks
Footage quotas
Freelancers
Full-time employees
Game engine
Generalists
Hand-drawn animated feature movie
Health risks
IK rigs
Inciting incident
Input capabilities
Internets
Intellectual property
Intranets
Leica reels
Lighting
Look development
Massively multiplayer online game, MMOG
Media asset management
Meetings
Milestones
Microcomputers
Modeling
Motion tests
Multicore CPU
Network, servers
Non-linear animation
Output capabilities
Operating systems
Parallel action
Pausing
Peripheral storage
Personnel
Placeholder geometry
Plan of action
Planning
Plug-ins
Postprocessing
Postproduction
Preproduction
Previsualization
Previz test
Processing power
Production
Production pipeline, strategy, team, technique, flow
Production-ready
Progressive complications
Proprietary software
Props
Proxy
RAID disk arrays
Remote collaboration
Rendering
Repetitive stress injuries, RSI
Resolution (story)
Resources
Resume
Review
Rigging
Rising action
Rough layout
Rushes
Scene layout
Schedule
Scratch dialog
Script
Shading
Shot assignments
Independent short
Software upgrades
Specialists
Standard length
Story development
Story genre
Story reel
Super-microcomputers
Supervision
Surprise
Suspense
T-1
Teamwork
Technical complexity
Technical animation
Technical director
Technical implementation
Technical notes, team
Television commercial
Texture painting
Turnkey software
Type of production
Upward compatibility
Visual effects
Visual style, look
Within budget
Written description
Written materials

(Characters by David Robles for *El Bosque Animado*. © 2001 Dygra Films.)

SECTION II

Modeling

(Previous page) Millions of polygons were used to model *The Last of the Leaves*, created with Zbrush, Maya, and Photoshop. (Image courtesy of Meats Meier.)

3.1.1 This surreal scene illustrates the concept of gluttony with an unusual assortment of objects and creatures created with multiple modeling techniques. (© Jim Ludtke.)

CHAPTER 3

Modeling Concepts

Summary

THE SCULPTING, SPATIAL DESCRIPTION, AND PLACEMENT of virtual three-dimensional objects, environments, and scenes with a computer system is called **modeling**. This chapter explores the basic concepts of the modeling process, including the numerical description of objects, moving and resizing objects in three-dimensional space, popular file formats, and some practical advice.

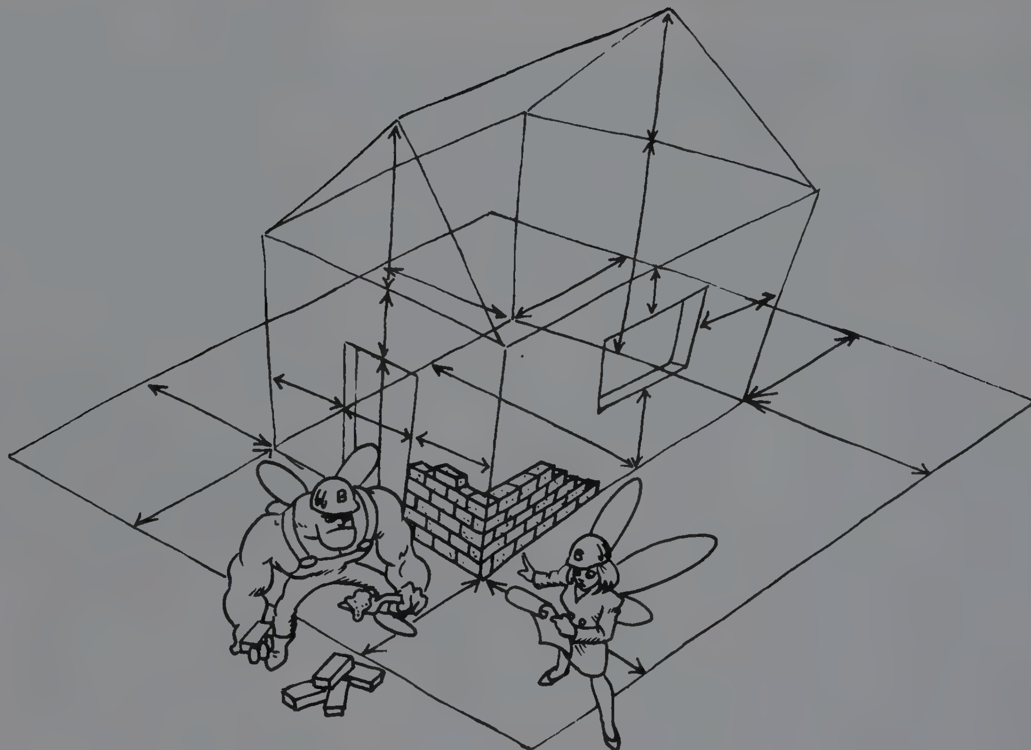
QUICK INDEX

3.1 Space, Objects, and Structures

We live in a three-dimensional world. We move among other people, climb mountains, run on the beach, and admire the landscape around us. We go in and out of buildings, walk up and down the stairs, drive through bridges, and grab utensils for writing, cooking, and grooming ourselves. Our daily life happens in three-dimensional environments and is full of three-dimensional objects and characters (Fig. 3.1.1). We see and feel three-dimensionality all the time. But unless we are involved in an activity or profession that is related to building things, like silverware or furniture, buildings or bridges, we rarely concern ourselves with how our three-dimensional reality is put together and what techniques are used for building it.

Modeling three-dimensional scenes with a computer program requires that we familiarize ourselves with the large selection of computer software tools used for modeling objects and environments. In three-dimensional computer modeling it is quite common to use a combination of tools for building just one object. For example, think of the difference between a chair that was built using just two tools—a manual saw and a hammer—and a chair that was built with six tools—a thick saw, a thin saw, a lathe, a curved chisel, a hammer, and a sanding tool. It is obvious that while the first chair could have an interesting design, the variety of shapes would be limited and directly related to the skill of the artisan. The second chair would have richer and more refined modeling. The simple computer-based modeling

(Top: Courtesy of Toru Kosaka, Studio EggMan.)



3.1.2 Building a simple rectangular room requires taking measurements that include the orientation of the walls, the location of the doorway and windows, and the distance between the walls. (Line drawing by Steve Rittler.)

tools are described in this chapter, and most of the advanced modeling tools are covered in Chapter 4. Now let's step back and talk about some general issues involved in modeling in three dimensions.

Many of the basic conventions used in three-dimensional modeling software describing three-dimensional scenes are based on the traditional conventions used in various disciplines. For example, to convey space in a clear and concise way, architects use conventions related to measuring, composition, and sequence. Even the design of a simple rectangular room requires measuring many times so that all the components of the room end up where they were planned to be (Fig. 3.1.2). Furthermore, to interpret accurately an architect's drawing and build it, masons need to take measurements. Over the ages masons and architects have developed conventions so that they can be precise and clear about measuring **spaces**, building **objects**, and arranging them in **structures**.

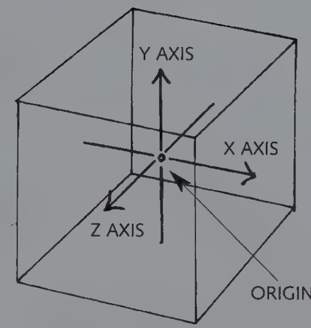
We use similar conventions to describe the dimension, placement, and sequence of objects and environments in a three-dimensional space simulated with a computer program. A beginning builder will soon find out that there are many different methods available for measuring space, and at first this variety can be confusing. But experienced builders usually find this richness of methods empowering and have the option of using one or another depending on the requirements of the project. Let's start our definition of three-dimensional space with the boundaries that define our **workspace**

or **scene**. The simplest way to do this is to imagine that we are working inside a large space like a cube or a sphere. We can think of this space as our world or environment. Objects that exist within the space are visible, those that fall outside are invisible (Fig. 7.2.1).

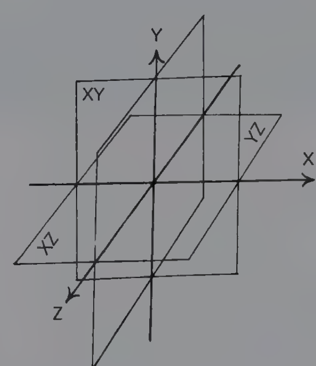
The main point of reference in this world is called the **world origin**. The origin is usually located in the center of the space, but it can also be placed or repositioned elsewhere depending on the modeling needs and strategies (Fig. 3.1.3). For example, if we were building a model of the solar system, it would make sense to have the world origin where the sun is, in the center, because all the other objects in the system are placed around the sun, and can be easily described in terms of the sun. If we were building an underwater scene that included both fish under the water and boats above the water, we might want to position the origin at the point where air and water meet. In the case of a three-dimensional model of an airport, the world origin could be placed at ground level, matching the position of the control tower. All three-dimensional spaces have three basic **dimensions**: width, height, and depth. A common method for representing these dimensions in a three-dimensional space is by using arrows or **axes** (Fig. 3.1.3). It is common to label the **axis** representing the width of a **three-dimensional space** with the letter X, the height axis with the letter Y, and the depth axis with the letter Z. The point in space where these three axes intersect, or cross each other, is the world origin.

The **rectangular coordinate system** can be used to define specific locations and accurately position the points of objects in three-dimensional space. René Descartes, an eighteenth-century French philosopher and mathematician, formalized the idea of using three axes labeled X, Y, and Z to represent the dimensions in three-dimensional space. The coordinate system he devised is commonly referred to as the **Cartesian** (or rectangular) **coordinate system**. Each axis in the system can be divided into many units of measurement. In principle these units are abstract values that can represent different units of measurement and scales of dimension. On each axis the values to one side of the origin are positive, and the values on the other side are negative. As shown in Figure 3.1.3, the positive direction of each axis in a right-handed coordinate system is represented with an arrowhead.

There are many ways of representing the direction of an axis and, consequently, the directions in which values on that axis are positive or negative. Usually though, in what is called a **right-handed coordinate system**, the values on the X axis become larger to the right of the origin, the values on the Y axis increase as they move above the origin, and the values on the Z axis grow as they get closer to us. In a **left-handed coordinate system** the values on the Z axis decrease as they get closer to us. There are several variations of the directionality of the rectangular coordinate system, but most three-dimensional modeling programs use the right-handed coordinate system to describe the virtual world.



3.1.3 The origin is a point of reference usually located in the center of the three-dimensional space. It can also be located elsewhere in three-dimensional space. A three-dimensional space has width, height, and depth dimensions each represented by the three axes in the Cartesian coordinate system. The numerical values in the figure correspond to those in a right-handed coordinate system.



3.1.4 The three planes, or views, that can be defined with the XY, XZ, and YZ pairs of axes are useful for building models from different points of view.

The three axes in the rectangular coordinate system can be paired with each other in three different ways so that each pair of axes defines a plane or a view. The XY axes define the **front plane**, the XZ axes define the **top plane**, and the YZ axes define the **side plane** (Fig. 3.1.4). There are other coordinate systems in addition to the popular rectangular coordinate system. The **spherical** or **azimuthal coordinate system** is also widely used because it provides a simple method for placing objects in a three-dimensional world in terms of their distance to the object, their angle around the point of interest, and their altitude angle above the point of interest. The spherical system is especially useful for placing and moving cameras and light sources in a three-dimensional scene (Fig. 3.4.8).

Any **world** or **global coordinate system** is useful for placing or moving objects in the world or in relation to each other. World coordinates are absolute values that are relative to the origin of the world. These coordinates do not depend on any specific object in the world and are applied to all objects in the world indistinctly. (Global transformations, as we shall read later in this chapter, are easily expressed in terms of the world coordinate system.) In addition to the world coordinate system, however, each object in the world can have its own **object** or **local coordinate system** (Figures 3.1.3 and 3.1.5). Object coordinate systems are values relative to the origin of the object, which is sometimes placed in the center of the object. For all practical purposes object coordinate systems are used only to specify positions, orientations, or transformations of the object in question. (Local transformations are almost always expressed in the coordinate system of the object.)

3.1.5 The world coordinate system is used to place and move objects, the entire building for example, in relation to the world origin. The objects' coordinate systems are useful for performing transformations only on specific objects, for example a single column. (Sainsbury's, view from Blackfriars Road South. © Hayes Davidson.)

3.2 Building with Numbers

Throughout time we have developed a sophisticated vocabulary for describing with the spoken or written word the shape of three-dimensional objects and their relative positions to one another. We can use that vocabulary for communicating with others about three-dimensional objects and their positions in space. But, even though

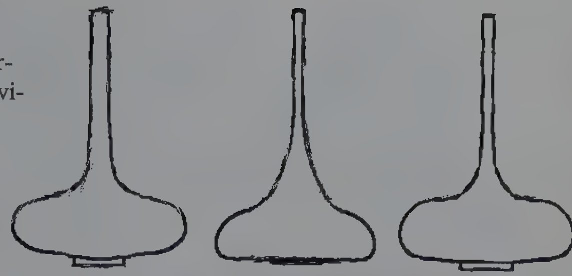
verbal descriptions of objects can be very concise, they lack precision. Verbal descriptions of objects can be interpreted in a variety of different ways, not only for the obvious issues of dimension but also for the subtle issues of proportion and shape. As you read the following description of a flower vase (Fig. 3.2.1) try visualizing it in your mind or drawing it with pencil and paper.

If you have some experience with modelmaking or pottery you were probably able to follow the description of the shapes in the vase and the relation between them, and your flower vase might look similar to the results shown in Figure 3.2.1. But if you have little experience with three-dimensional models, your sketch may be quite different, or you may have been unable to finish reading all of the description. Maybe you lost interest because you found it difficult to visualize all the shapes and the ways in which they were attached to one another.

Most individuals and today's computer systems are incapable of recreating in detail verbal descriptions of complex three-dimensional shapes. Computer modeling in three dimensions requires precise and unequivocal descriptions of shape. Using numbers is the method of choice for precise and unequivocal descriptions of shapes and their location in space. With numerical description we can specify the position of an object in space and the details of its shape: height, width, depth, diameter, curvature, and number of sides. Figure 3.2.2 illustrates most of the numbers required for describing a simple three-dimensional shape.

Much of the success in modeling (and rendering) three-dimensional objects and environments with a computer system lies in understanding how a particular computer system describes a shape with numbers. The exact numbers can mean different things to different software programs. For example, some systems give great attention to the decimal, or floating point, numbers (e.g., 5.379) that describe the subtle shape of a small curve, while other computer systems may ignore those numbers completely and recreate the curve based on a whole numerical value (e.g., 5). We must also keep in mind that some of the numbers that describe a shape might be of little value to us if we were building the object with traditional materials, such as wood, but the same numbers often provide the computer system with crucial information for building the object with computer modeling techniques. For example, the order in which we number points in a shape can yield very different results.

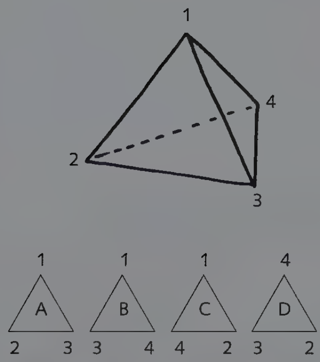
The essence of all software-based three-dimensional modeling techniques consists of creating a **data file**, or a list of numbers, which defines models in a way that can be understood by the computer program. Whether we are creating a simple cube or a collection of computer shapes representing a human hand, the numbers describing the object are kept in a file so that the program can load them into memory, display them, modify them, display them again, save them, and so on. The files that contain the data describing the



Verbal Description of a Vase

1. The vase has a long neck and a short, round base.
2. The neck is about five times the height of the base, and the width of the base is about twice its own height.
3. The cylindrical neck grows out of the base slowly.
4. At the point where the neck touches the base it has a width that equals the height of the base.
5. The neck narrows as it moves upward and, as it passes the first fifth of its height, it reaches a thin delicate width that remains constant from there on.
6. A small section of the oval shape that constitutes the base of the vase is sliced off so that the bottom of the base is flat.
7. The resulting sharp edge at the bottom of the base is rounded off just a little bit.
8 Halfway between the edge of the base and its center, a thin slice of a short cylinder is attached to the base.

3.2.1 Different interpretations of the verbal description of a flower vase.



Geometry Format #1

Vertex	X	Y	Z
1	0	0	0
2	-1	-2	1
3	1	-2	1
4	0	-2	-1

Facet	Vertex	Vertex	Vertex
A	1	2	3
B	1	3	4
C	1	4	2
D	3	4	2

Geometry Format #2

Facet	XYZ	XYZ	XYZ
1	0 0 0	-1 -2 1	1 -2 1
2	0 0 0	1 -2 1	0 -2 -1
3	0 0 0	0 -2 -1	-1 -2 1
4	1 -2 1	0 -2 -1	-1 -2 1

3.2.2 The difference between two geometry formats for the same object is quite evident here. These two listings were generated with two different types of modeling software.

object are called **geometry files**. Examples of all geometry file formats are discussed later in this chapter.

In most modeling systems, three-dimensional objects can be modeled by typing the numbers that describe the object directly into the system. This method of **direct numerical description** can be quite tedious and time-consuming. We rarely use it unless we are looking for an extremely specific shape or a detail that can be hard to model with regular modeling tools. Even when we use the interactive modeling tools provided by the software it is still possible to peek at the numerical information that the software uses to describe and manipulate three-dimensional shapes. Most systems allow us to get this numerical information in varying degrees of detail (Fig. 3.2.2). Both the shape of an object and its position in three-dimensional space are expressed in terms of numerical values (Fig. 4.6.1). From the computer's point of view, numerical values are easy to manipulate and easy to repeat. This facilitates the building of three-dimensional objects with modeling software, as well as the duplication of objects using the **cut-copy-paste** techniques used by most of today's programs.

3.3 Vertices, Edges, and Facets

Now that you have learned how to locate points in three-dimensional space and how to create and edit lists of numbers that represent XYZ spaces you are ready to start thinking about building a simple model. The three-dimensional object illustrated in Figure 3.3.1 is defined by four points, six lines or edges, and four planes or facets.

Points, lines, and facets are among the basic elements that can be used to build three-dimensional objects. A **point** can be easily defined by its XYZ location. A **line** can be defined by the XYZ location of its two endpoints. In the context of three-dimensional geometry a corner point is also referred to as a **vertex**, which is defined by the intersection of two or more edges. An **edge** is defined by two adjacent surfaces. A **facet**—as those in a cut diamond—is a planar surface that is defined by the position of its bounding lines. A three-dimensional object is usually composed of several points, lines, and facets, and it can be described in software with a list of numbers. This list is usually generated automatically by the computer program, but it can also be generated directly by the user. As stated earlier, it is not necessary in most cases to input all of these numbers by hand. In fact, we do not need to be aware that all this number-shuffling is taking place. But once in a while you will encounter modeling situations when it will be paramount that you understand the meaning and proper structure of these numbers. It is for those occasions when the information in this section will come in handy.

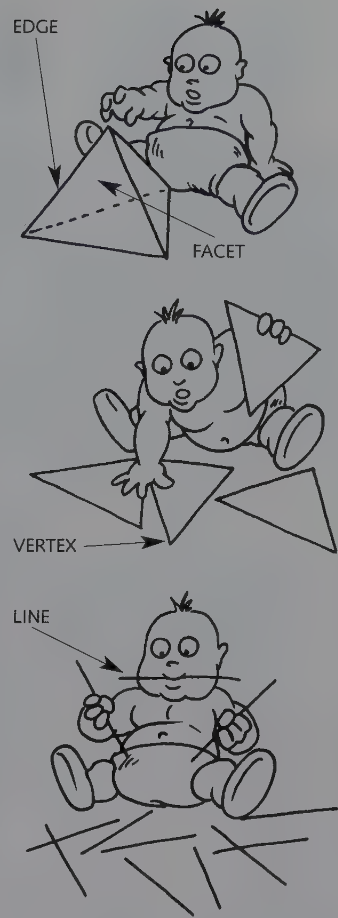
Simple objects, like the one pictured in Figures 3.2.2 and 3.3.1, and even more complex objects, can be easily described or edited in most modeling software providing their point XYZ positions and connectivity to the three-dimensional program. Whether this is done

by typing their numerical values directly on the keyboard or by transferring the XYZ data collected by a three-dimensional scanner, a simple methodology can be followed. First, label all the points and all planes in your sketch or printout—if one is available (Fig. 3.2.2, top). Then write down the XYZ position of each of the points in list form (Fig. 3.2.2, Format #1 points). Finally, make another list that includes each of the planes and all the points that must be connected to define them. It is important that all the points in each plane are connected in the same direction—either clockwise or counterclockwise (Fig. 3.2.2, Format #1 facets). Some computer programs require that you connect the points clockwise, while others require counterclockwise, but usually all programs require that the order be consistent throughout describing the entire object.

The facets or **planar surfaces** that define most three-dimensional objects are also called polygons. The word **polygon** has its roots in the Greek word *polygónon*, which means "with many angles." Polygons are closed planes bounded by straight lines. They can be regular or irregular, and can be used to create three-dimensional objects called polyhedra. Each polygon in a polyhedron (singular for polyhedra) has a surface normal, which is a vector perpendicular to the surface (Fig. 9.1.2). As described in Chapter 9 surface normals play a central role in determining whether surfaces on polyhedra are visible or hidden, and whether they should be shaded or not.

In addition to regular polyhedra (Fig. 4.3.1), polygons can also be used to build polygonal meshes that describe simple and complex objects. A simple teacup model, for example, may require less than a hundred polygons, while a detailed teacup would require several hundreds of polygons. Complex objects, such as the detailed model of a human, may require thousands of polygons. Figure 3.3.2 shows polygonal meshes in **wireframe rendering** style, with the hidden surfaces removed. The modeling of natural phenomena, such as a forest or simulation of the explosion of a supernova star, would require millions of polygons.

We can also define objects with curves instead of straight lines, and curved surfaces instead of flat polygonal surfaces (more in Chapters 4 and 5). Unless otherwise specified, most modeling techniques described in the three modeling chapters focus on building surfaces that are hollow shells—this is called boundary geometry. More on this and CADAM's different approach later in this chapter.

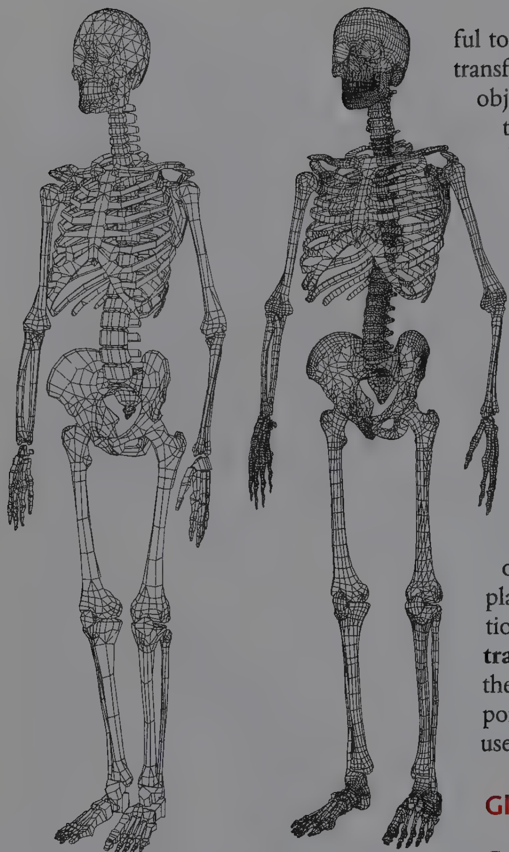


3.3.1 Three-dimensional objects are defined by points, lines, and planes. This simple pyramid has a total of four vertices and four sides.

3.4 Moving Things Around

Once we have built some objects we can move them around in three-dimensional space and create a composition or a scene. Sometimes it becomes necessary to move some of an object's components—a group of points, for example—before the modeling is completed.

The functions used for modifying the shape of objects, their size and proportions as well as their position in space, are called **geometric transformations**. The name of these simple but power-



3.3.2 The first skeleton model is built with 8,979 polygons, the second one with 35,305, and the third one (opposite page) with 141,788 polygons. Notice the higher density of polygons in areas of the surface that have more modeling detail. (© Viewpoint Datalabs, used with permission.)

ful tools obviously comes from the fact that they can be used to transform—to change, to move, to modify—the geometry of objects. In effect, these **mathematical operations** can modify the numerical information that describes the objects that we build in the environment and even the environment itself. The most widely used geometric transformations are translation, rotation, scaling, and perspective projection.

Geometric transformations can also be applied to the camera that "looks" at the scenes we model and arrange, and also to the lights that reveal our creations to the camera. (See Chapter 7 for more information on camera motion, and Chapter 8 for more on moving lights around.) In general, when specifying transformations to be applied to a single object or a group of objects it is important to specify the type of transformation, the axis or axes on which the transformation is to take place, the point around which the rotation or series of rotations will occur (whether the transformation is local or global), and the order in which transformations are to take place in a sequence of several of them. Geometric transformations are usually calculated by most programs with the aid of a **transformation matrix**. This 4×4 matrix is used to calculate the new XYZ values after a transformation is applied to all the points of a three-dimensional element. A few programs allow users to manipulate XYZ values directly in the matrix.

Global or Local Transformations

Geometric transformations can be performed on single objects or on entire environments. Transformations that are applied to the objects using the environment's axes and/or origin are called **global transformations**. When transformations are applied to a single object—or a limited selection of objects—using the object's own axes and origin, they are called **local transformations**.

We can start to define a local transformation by selecting or activating one or several objects. In general, when one object is selected as the recipient of a local transformation, the object's center and axis are used as the centers of rotation and scaling, and the axis of translation, rotation, and scaling. (The centers of rotation or scaling are usually located in the center of the object unless specified otherwise.) Some programs, however, offer the option to apply local transformations to an object based on the environment's center and/or axis instead of the object's center and/or axis. The results can be quite different (Fig. 3.4.1). For example, an object scaled along its axis after being rotated retains its shape, while an object scaled along the world's axes will not retain its shape. Check the manuals of the software you use to find out how local and global transformations are implemented. Having a clear understanding of this is crucial to the correct operation of your software.

When performing global transformations, or local transformations that occur along or around the global axis, the order in which transformations are applied to an object or a series of objects can affect the final result. For this reason rotation and scaling sequences should be planned carefully, although translation sequences can be applied in any order. **Concatenated transformations** is the name sometimes given to a series of global transformations applied in sequence. Figure 3.4.2 illustrates the different results obtained by applying the same global transformations to a trio of objects but in a different sequence each time.

In general, if all objects in a scene are active, the transformation is global and applied to all objects. Most software programs apply transformations to all the active objects. When performing a global rotation or global scaling, the center of the environment usually doubles as the center of rotation and scaling for all the objects unless specified otherwise.

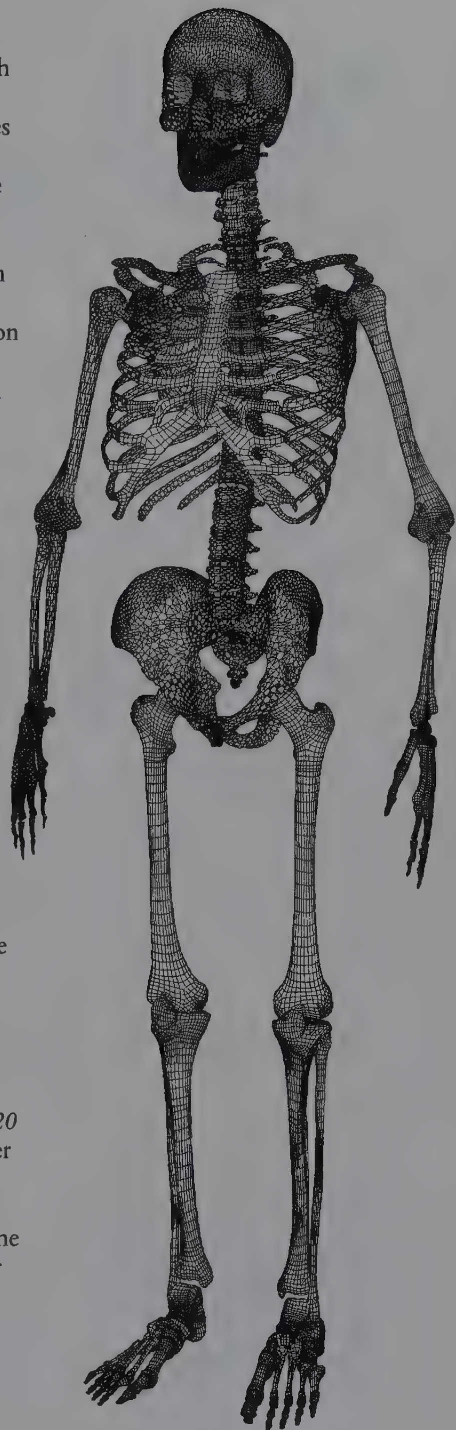
Absolute or Relative Values

When working with most interactive modeling programs, applying transformations to one or several objects is as easy as selecting them and dragging them to a new location in three-dimensional space. It is quite common to use the mouse and the mouse button to control the position, orientation, and size of the models in the environment. However, it is sometimes necessary to type specific values on the keyboard for controlling the exact position, orientation, and size of models. When typing numerical values becomes a necessity one must keep in mind that all transformations can be specified as **absolute values** or as **relative values**.

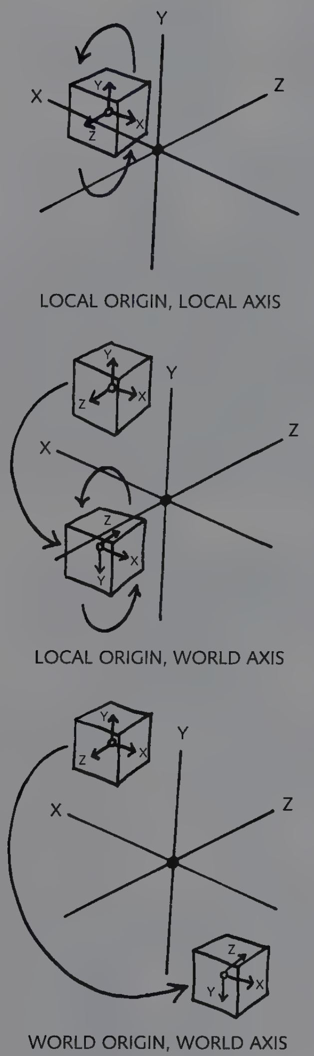
Absolute values, or numbers, always refer to an exact position in space where the object must be relocated regardless of where the object was located in space before the transformation. Relative values, as their name indicates, are numerical values that express the number of units that must be added or subtracted to the current position of the object. Relative values are relative to an existing absolute position. For example, if we have a sphere with a center located at XYZ coordinates 30 30 30, the command *trans sphere 0 20 0* (if the numbers were relative) would reposition the sphere's center at XYZ coordinates 30 50 30 because 20 units would have been *added* to the sphere's position. However, if the numbers being used were absolute numbers, the sphere's center would be relocated to the 0 20 0 XYZ position, regardless of the fact that the sphere's center was previously located at 30 30 30.

Translation

Translation is the simplest of all geometric transformations. This operation is used to move an object or group of objects in a linear



141,788 POLYGONS



3.4.1 The position of an object's center has important implications in the results of the transformations applied to it. The first cube has been rotated locally around its own X axis and its own origin. The second cube has been rotated around the world's X axis and its own local origin. The third cube has been rotated globally around the world's X axis and origin.

way to a new location in three-dimensional space (Fig. 3.4.3). Translation is the simplest and easiest to control of all geometric transformations. Translation can occur along one axis or along several axes at the same time. The order in which several global and local translations are applied to one object does not affect the final position of the object. For example, an object that is translated 5 units along the X axis, then 10 units on the Y axis, and finally –7 units on the Z axis would end up in the same positions as an object that is translated first 10 units along the Y axis, then –7 units on the Z axis, and finally 5 units on the X axis.

Rotation

Rotation is the geometric transformation used to move an element or group of elements around a specific center and axis. The amount of rotation is usually specified in terms of an angle of rotation (measured in degrees) and a direction of rotation (Fig. 3.4.4).

Depending on whether the rotation is global or local, objects can be rotated around their own center, the center of the environment, or even the center of their "parent" in a **hierarchy** of objects (for more on transformation of model hierarchies see Chapters 5 and 11). When rotating an object around its own center it is possible with many programs to reposition that center. Consequently, the center of rotation of an object may not always be placed in the geometric center of the object.

Rotations can be used to present different sides of an object to the camera. Rotations are effective for arranging subtle details in a scene, such as to expose sides of a model with the most interesting shapes or details, to simulate motion, or to emphasize the perspective of the objects in the scene.

Because rotations always happen around an axis, it is important to know which way rotations are supposed to occur. Depending on the value (positive or negative) that defines a rotation, the rotation can be clockwise or counterclockwise. In a right-handed coordinate system, positive rotations are always counterclockwise and negative rotations are clockwise. A simple method for remembering the direction of rotations consists of representing the axes on which the rotation is taking place with our extended right-hand thumb, as shown in Figure 3.4.5. If the thumb points to the positive direction of the axes, the direction of a positive rotation is defined by the direction in which we close the hand and make a fist.

Scaling

Scaling is a geometric transformation used to change the size and/or the proportion of an element or a group of elements. Scaling can be applied to an object in a proportional or a nonproportional mode. **Proportional scaling** consists of resizing an object along each axis in equal amounts. The result of proportional scaling is always a larger

or smaller object with the same proportions as the original object. With **nonproportional scaling** the object may be resized by different factors along each axis. Nonproportional scaling can be used to change the proportions of a three-dimensional object so that it becomes taller or shorter, wider or narrower, or deeper or shallower (Figs. 3.4.6 and 3.4.7). Because of its ability to easily modify the shape of objects, nonproportional scaling is widely used in computer animation to simulate the "squash and stretch" distortions typical of cartoon - style animation.

When a scaling operation is performed, not on a single object but on all the objects in the environment, we get an effect that is reminiscent of a camera zoom.

Perspective Projection

Perspective projection is a transformation of critical importance because it makes possible the representation of three-dimensional environments on the flat surface of the computer's monitor or a sheet of paper. A perspective view of a three-dimensional scene is created by projecting each point of an object from the viewpoint onto the image plane (Fig. 7.2.1). The points in the three-dimensional object coordinate system are then transformed to the two-dimensional image coordinate system.

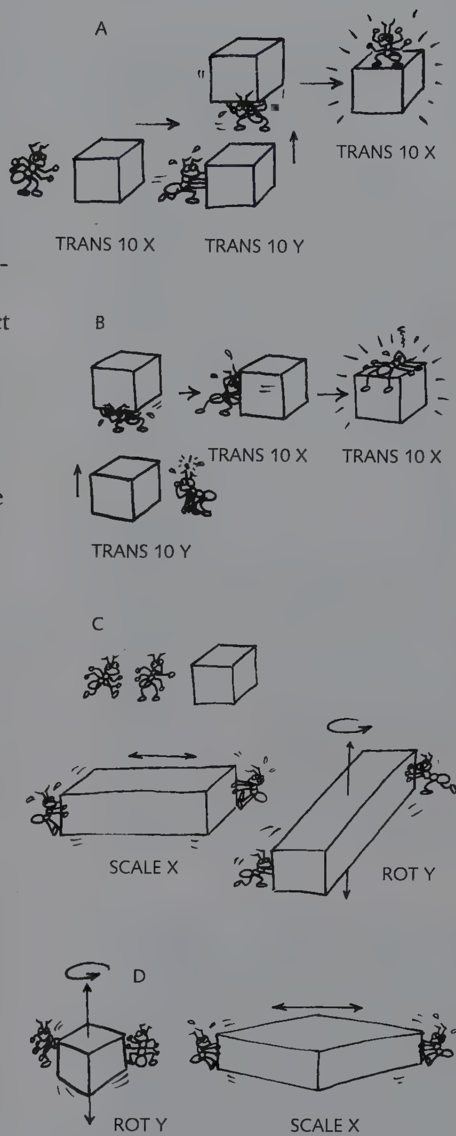
Perspective projection is a transformation that happens automatically in virtually all three-dimensional software. It is not necessary that we ask for perspective projection each time we do something to our scene. The three-dimensional environment is constantly being transformed into a two-dimensional view using perspective projection techniques. See Chapter 7 for more information on perspective projection.

Navigation

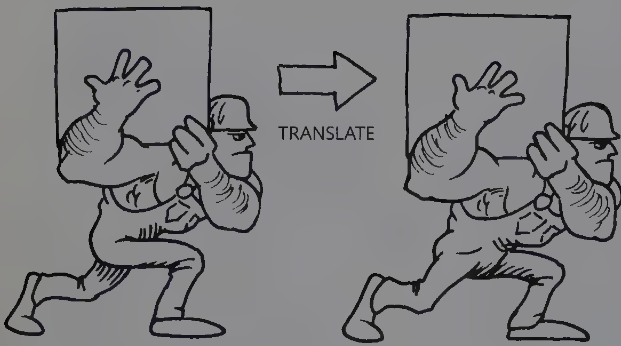
Navigation usually refers to the motions that place the camera in different parts of the scene. Navigation can be used during the modeling process for looking at the models from points of view that show the model in more detail. Navigation also takes place during the layout process when the camera is positioned to focus on areas of interest or where it helps tell a story more effectively.

The spherical or azimuthal coordinate system is often used to navigate through the world by specifying camera positions in terms of the camera's angle around the horizon, its angle above the horizon, and its distance from the object (Fig. 3.4.8).

What a virtual camera sees is defined by the camera position, its point of interest, and the camera lens. These characteristics can be quickly set by typing numerical values on the keyboard, or interactively by directly manipulating the camera with a variety of input peripherals that include the mouse, graphics tablet, trackball, joystick,



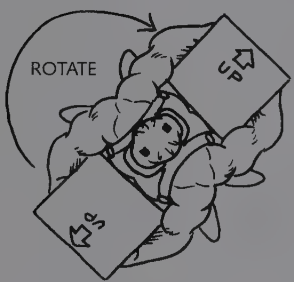
3.4.2 The first two examples (A and B) show the same translations applied to the same object in two different sequences. The results are identical; in both cases the object ends up in the same place. Examples C and D show two different resulting shapes after applying the same rotation and scaling to an object, but in a different sequence.



3.4.3 An object being translated.

and dials. Even though all camera positions and moves can be input from the keyboard, it is more practical and fun to position and move the camera interactively. In any case, each possible camera move will result in the modification of at least one of the camera's three basic values: position, orientation, and focal length.

The motions of computer animation virtual cameras are based on the camera moves defined in traditional cinematography. Most programs use the same camera names used in traditional cinematography, but some use a different nomenclature. All of the camera moves, even the most complex ones, can be expressed in terms of translations or rotations around one or several camera axes (Fig. 3.4.9). A **dolly** is a translation of the camera along its X axis, a **boom** is a translation along its Y axis, and a **truck** is a translation along its Z axis. A **tilt** is a rotation of the camera around its X axis, a **roll** is a rotation around its Z axis, and a **pan** is a rotation around its Y axis. Sometimes a tilt is called a **pitch** (as in airplanes pitching), and a pan is called a **yaw**. A **zoom** is a special type of camera move where only the camera's simulated focal length is modified but its position and orientation remain untouched.



3.4.4 A bird's-eye view of an object being rotated.

All the camera motions described here can take place in the perspective window, but in some programs some camera motions—such as yaw/pitch or azimuth/elevation—cannot be fully appreciated in the orthographic views because these motions can only be fully represented in three-dimensional space, and not on a flat surface. (Read Chapter 7 for additional information on cameras and camera moves.)

3.5 File Formats for Modeling

There are many formats for saving the information contained in three-dimensional geometry files. Many of the existing **file formats** containing descriptions of object geometry are exclusive to specific computer programs and are not portable. The information contained in a **native file format** is optimized and formatted according to conventions that are particular to the software in question, and the files are usually not compatible with other programs. A few geometry file formats are **portable**, which means that they can be exchanged among several programs.

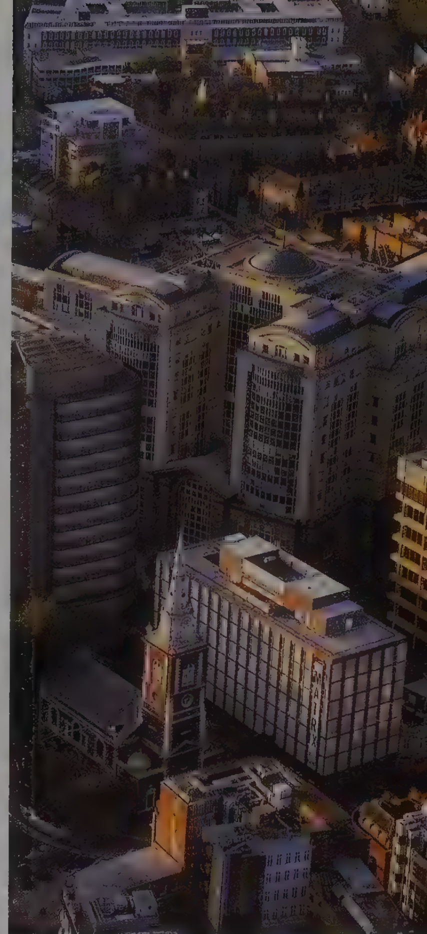
3.4.5 In a right-handed coordinate system the direction in which your hand closes to make a fist is the direction of a positive rotation around any axis, represented by the extended right-hand thumb.

The obvious advantage of using native, or custom, file formats is that it is easy and fast for any particular program to read files in its own native format. Files saved in native formats usually load faster and require less space for storage. There are a number of file **conversion utilities** that can translate geometry files in native formats between applications in varying degrees of accuracy. Models that have been built with standard techniques or that have a simple topology can usually be converted in this way successfully (Fig. 3.5.1). But trying to convert complex modeling files from one native

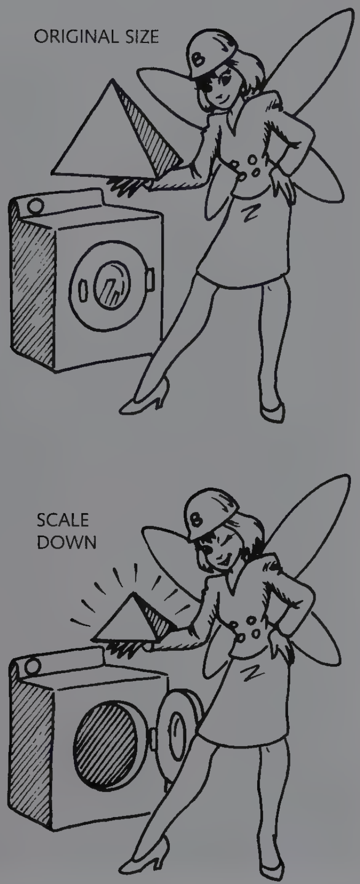
file format to another might modify some subtle details—or destroy them altogether—and might require a fair amount of manual adjusting. Solutions to the format incompatibility problem include using "universal" file formats for saving information about three-dimensional models, or converting one native file format directly to the native file format of another program.

The file formats used for transporting geometry information between modeling programs are often called **universal file formats**, and two of the most popular ones include OBJ and DXF. Two other formats, X3D and VRML, are commonly used on Web applications. The **OBJ** format, or **.obj**—short for object—was originally developed by Wavefront Technologies and popularized by Alias | Wavefront computer animation software. OBJ files essentially contain geometry information about the position and normals of vertices, associated image maps, and the object's facets. OBJ files can also be linked with other types of files that contain additional or related information, such as materials and shading parameters. The **DXF** format, short for **Drawing Interchange Format**, was developed by Autodesk, Inc. for handling both two- and three-dimensional geometry information. The DXF format is also known as **Drawing Exchange Format**, and is widely used in computer-aided design (CAD) applications. Even when using universal file formats to save three-dimensional information, there can be minor differences in the ways different programs interpret the information. This latitude in interpretation is due to the fact that many of the universal file formats describe three-dimensional information in a general way. The DXF files, for example, contain some two-dimensional information that is often discarded when imported by three-dimensional software. It is also common for three-dimensional programs to interpret in different ways the precision and/or the curvature of a line that defines a surface.

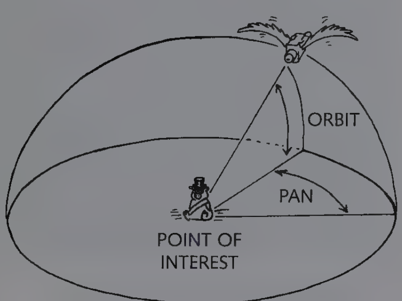
Virtually all three-dimensional modeling programs offer some degree of **file conversion.** That capability is found either inside the standard file management options (under a command or menu option such as Import or Retrieve) or as a stand-alone utility conversion program that can be executed outside of the modeling program. Most programs can also **export** their geometry data into other native or universal file formats. Many of today's three-dimensional modeling programs also offer some degree of **foreign-to-native** file format conversions. The number of data formats that a particular three-dimensional program may be able to convert may range from just a couple to several dozen formats. All file format conversions are controlled by **import filters,** which are tables that instruct the conversion utility program how to translate each of the elements encountered in the original—or foreign—file. Figure 3.5.1 shows how a particular file conversion program exports a data file into three different formats. Even when the most reliable file formats and conversion filters are used there are always small variations between the results obtained with different programs. A wide range of options is avail-



3.4.6 These buildings and tunnels were created by duplicating and scaling a variety of shapes and an extruded arc. (Top: Aerial view of St. Botolph's House. © Hayes Davidson. Bottom: © Jim Ludtke.)



3.4.7 An object being scaled, in this case by using a shrinking machine.



3.4.8 The spherical or azimuthal coordinate system allows for orbiting or panning the camera around the subject.

able in software when importing or exporting data in the DXF format, for example: constructing 3D solids, joining adjacent coplanar facets, making cardinal curves from splines, exporting facets as polylines, including spline and Bezier controls, subdividing concave facets, connecting holes to facet edges, or triangulating facets. One of the reasons for offering so many options in a conversion to a "standard" format is because not all the aspects—or options—of the DXF format are supported by all programs that are capable of reading DXF files. Figure 3.5.2 illustrates how a specific program deals with one aspect of the native-to-DXF file format conversion.

The results obtained with different file conversion utilities vary widely. Some file conversions are almost flawless (with only minor details requiring adjustment), while others rarely produce desirable results. There is no easy way to know if a file conversion program will work perfectly or not: each has to be tried and evaluated.

The **Virtual Reality Modeling Language**, widely known as **VRML**, gained popularity during the early 1990s as a convenient way to describe, in a portable format, three-dimensional environments for real-time online display (Fig. 3.5.3). VRML is also known for its **.wrl** extension. One of VRML's main innovations was to allow for the creation of virtual reality environments where multiple participants can interact with each other in real-time three-dimensional spaces. VRML is still used today, but X3D is a newer open-source standard for implementing interactive three-dimensional environments on the Web and in embedded devices. **X3D**, for Extensible 3D, is a **scene description language** that addresses object geometry, real-time rendering, navigation, and interaction. X3D also supports networking of virtual environments, NURBS support, and humanoid biped animation based on sophisticated rigging and animation variables such as a joint and end-effector hierarchy plus deformable skin controlled by weighting. Visit www.web3d.org, and see the end of Chapter 4 for additional aspects of real-time geometry (Figs. 4.7.1–4.7.10) , and Chapter 6 for real-time rendering.

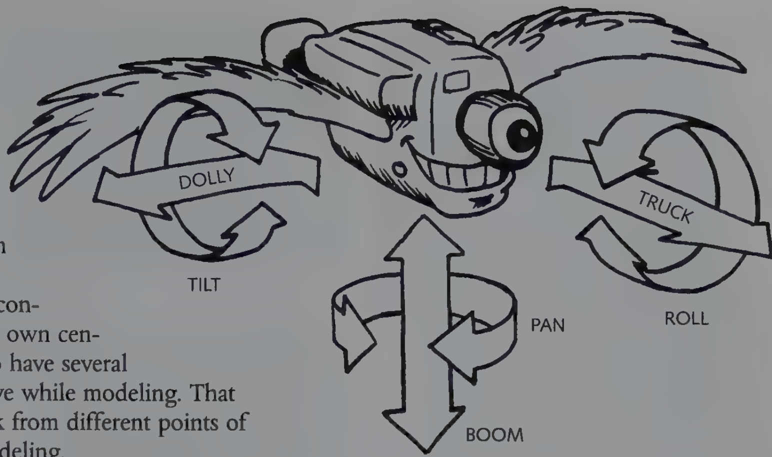
3.6 Getting Ready

Modeling can be a time-consuming activity because of the great attention to detail that is required. In spite of the flexibility offered by most computer modeling systems, trying to fix complications caused by poor planning sometimes can be more time-consuming and headache-inducing than starting over again. For this and other reasons—also related to practical issues such as time and budgets—consider some of the preproduction guidelines listed below.

Use Multiple Camera Views While Modeling

It is often quite advantageous to model a three-dimensional object with multiple camera views. A sculptor, for example, walks around the sculpture as he works on it in order to have a clear mental picture of

how all the parts and shapes relate to one another. As soon as some of the shapes change, others require reshaping and fine-tuning. In much the same way, it is convenient for an individual using a computer three-dimensional modeling system to look at the object from multiple points of view as it evolves. This can be easily accomplished by constantly rotating the object around its own center. But it can be more convenient to have several views—often called windows—active while modeling. That way one can get immediate feedback from different points of view while concentrating on the modeling.



3.4.9 Navigating through three-dimensional space can be done by using the traditional cinematography camera moves: dolly, boom, truck, and rotations: tilt, pan, and roll.

Most three-dimensional modeling systems allow you to have four active views open at once (Fig. 3.6.1). It is common to use a front view, a side view, a top view, and a camera view. The camera view usually allows for total control of the point of view: the simulated camera can be placed close to the object being modeled—for examining details—or far away, for evaluating the overall shape. Some viewing positions such as 60-0-30, 45-0-45, 20-0-120, 45-0-220, and 30-0-60 are popular for positioning cameras during the modeling process because these positions resemble the angles of some of the standard three-dimensional projections commonly used in drafting. Using multiple views may be impractical when working with huge scene files and very complex geometries. Sometimes it takes too long to process this information and to update the screen. This may force you to work with a single camera view and switch between views as needed. (Read Chapter 7 for more information on setting the camera.)

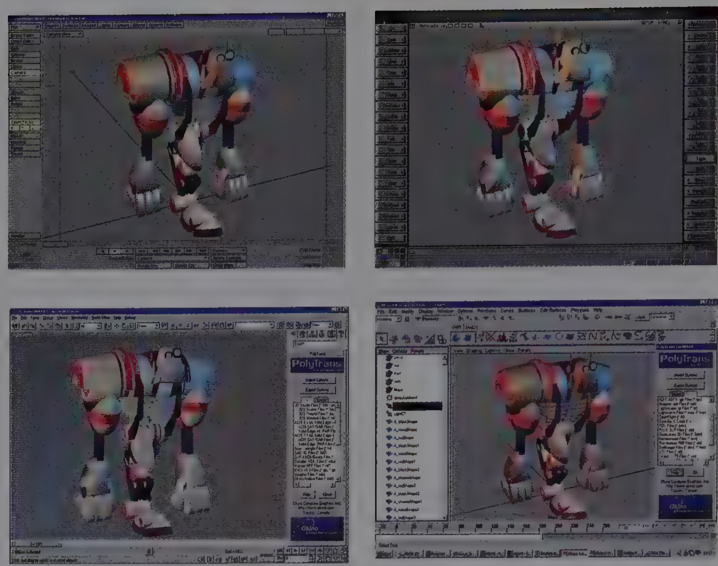
Do Not Lose the Blueprints

Blueprints are a necessity in cases where the objects to be modeled are too complex and detailed for improvisation and memory. In those instances, it becomes paramount to hold on to the original blueprints (Fig. 3.6.2). Even after you think the models are finished, you or someone else on your team (or on the opposite team) may decide that the models are not finished after all. In such an event, you or someone else may need the blueprints again.

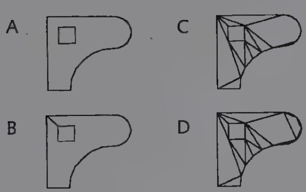
Sketch Your Ideas First

Sketching your modeling ideas on paper or modeling them in clay can sometimes be faster and more economical than starting to model directly on the computer. While there is nothing intrinsically negative about creating three-dimensional models without having a sketch at hand, in most cases starting that way increases the chances

(Zinkia™ and Pocoyo™ © 2005 Zinkia Entertainment SL.)



3.5.1 Four images show how a geometry file originally created with LightWave™ is converted with PolyTrans software into three other native file formats: 3D Max, Softimage, and Maya. (ApeBot model © 1999 Matt McDonald, Vision Scape Imaging, and NewTek, Inc. Respective screen shots © 1999 NewTek, Inc., Kinetix, Inc., Avid Corp., and MultiGen-Paradigm Corp. Images courtesy of Okino Computer Graphics.)



3.5.2 An example of the decomposition process of a concave shape with a hole (A) when exported to the DXF file format. First all holes are connected to the edges of the shape (B), then the concave shape is decomposed into several convex parts (C); next all parts are further decomposed into four-sided parts (D), and finally, the shape is triangulated so that all the component parts become elements in a triangular polygonal mesh. (Courtesy of auto•des•sys, Inc.)

of running into small problems that might have been easy to avoid. Small details such as the ways in which two complex shapes will blend into one another, for example, can be visualized faster and cheaper on paper or modeling clay. In most cases, it is not an issue of whether something can be sketched and visualized directly on the computer, but one of economics: It can be much more expensive to sketch with the computer system than with a soft graphite 2B pencil, for example, plain white paper, and an eraser.

Another advantage of developing three-dimensional ideas on paper or with modeling clay is that both media are absolutely portable and do not present any type of compatibility problem. This is especially important when you are required to present your work to others before actual production starts. It is very easy to show someone a sketch on paper; that can be done anytime and anywhere. There is no need, for example, for your client to travel to your location because they do not have computers or for you to have to reserve one of the workstations in your company so that your client can see your ideas. There are few techniques for sharing your visual ideas with others as direct, portable, participatory, and friendly as a drawing on a piece of paper or a rough clay model.

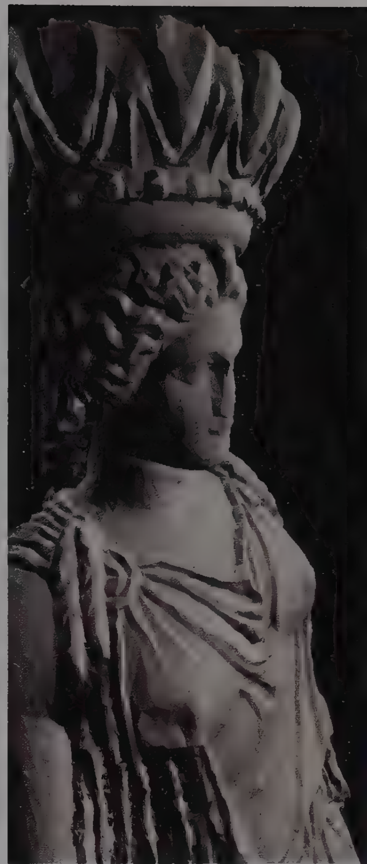
If you are still unconvinced, keep in mind that a sketch is just a quick study, a rough drawing, a draft, a preliminary outline. A sketch is not a polished rendering or detailed sculpture. A sketch for a three-dimensional model should take just a few minutes to complete; it should be tentative but also offer enough detail in the more complex areas of the object. Sketches are meant to be aides in the production process, not works of art to be framed and admired by museum-goers. In some instances, it can be useful to complement the drawn sketches with short, explanatory notes regarding details

such as the proposed modeling technique or the number of polygons needed to define the curvature of a part, or the advantages and disadvantages of the solution proposed in the sketch (Fig. 3.6.3).

Polygons or Curves?

A few of today's three-dimensional modeling programs are exclusively polygon-based, but most offer both polygons and curves. Choosing between polygons or curves for modeling a three-dimensional object has obvious implications as far as the model's shape is concerned (for more on these implications see "Geometric Primitives" in Chapter 4). However, the rendering implications of modeling with polygons or curves are less obvious but critical in some cases (Figs. 3.6.4 and 3.6.5).

As we will learn later in Section III: Rendering, many rendering programs require polygonal structures. This means that whenever curves are used—but before the three-dimensional model can be processed by most rendering programs—the curves have to be converted to polygonal structures. In most cases, this conversion is not a problem: many programs perform this conversion automatically. The technique of subdivision surfaces (discussed in Chapter 5) offers a convenient "best of both worlds" compromise. However, in some cases, whether to start modeling with polygons or curves can become an issue that requires planning. For example, some curve-based modeling programs do not accept polygonal models, and likewise, some polygon-based modeling programs will have a difficult time reading files of models that have been specified with curves. Furthermore, in many three-dimensional modeling programs—even those that offer both polygonal and spline-based modeling techniques—some advanced functions such as bevelling or clipping will work only on polygonal meshes. Oftentimes the software that offers two-way conversions between polygon meshes and spline surfaces does so at the expense of the shapes involved. When these conversions are performed there is often some distortion that might require time-consuming, point-by-point rearranging of the results.

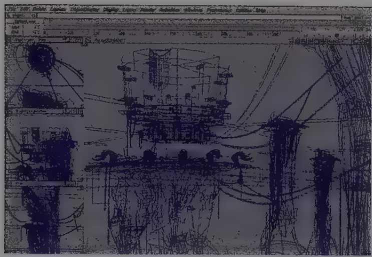


3.5.3 A 3D VRML model of one of the six caryatids, column in the shape of a sculpted female figure, in the *Erechtheion*—a building across from the *Parthenon* in the Athenian Acropolis. The data was digitally scanned from a physical model with a custom-built light scanner. This VRML version has a tenth of the polygons in the high resolution version. (Image from the online Parthenon Sculpture Gallery. © 2008 USC Institute for Creative Technologies, Basel Skulpturhalle, and VCG CNR-Pisa.)

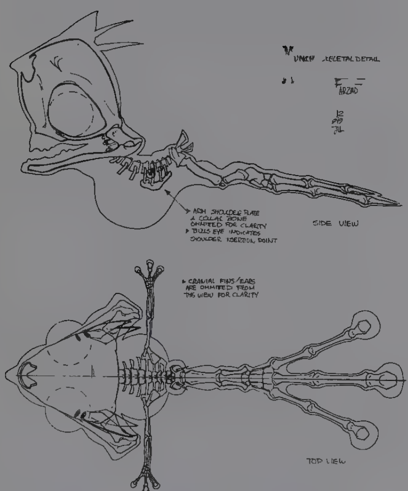
Will the Model Be Used for CADAM?

Objects that are modeled just to be rendered or animated are built differently from objects that will serve as models for **computer-aided design and manufacturing** (CADAM). It is extremely important to know whether one's models will be used for CADAM. If so, a specific modeling methodology has to be chosen and followed consistently throughout the project. Figure 3.6.6 shows a three-dimensional model that was used to fabricate a sculpture with stereo lithography techniques (Fig. 15.8.4).

The two significant differences between modeling for CADAM projects or modeling for animation projects lie in the modeling technology used, and in the fabrication and structural implications of the objects modeled. Few computer systems offer both modeling tech-



3.6.1 Most three-dimensional programs can display simultaneously multiple projections of the camera view. Shown here are the perspective projection, and the top, front, and side orthographic projections. See the sketches for this environment in Figures 2.4.2 and 2.4.3. (© 1999 Oddworld Inhabitants, Inc. All rights reserved.)



3.6.2 Blueprint detailing surface shape, proportions, and skeleton of a character. (© 2003 Oddworld Inhabitants, Inc. All rights reserved.)

niques. The majority of software is either just boundary-based or solid-based. In those situations this issue is automatically solved by the limitations of the software. It is in cases when the software has both capabilities that we have to choose between object shells or solid objects. When we build three-dimensional objects for rendering or animation purposes, we are almost always interested in the surfaces of the objects and rarely in their inside volume. For that reason, when we model objects for rendering and animation, we usually use **boundary** and **geometry modeling techniques**. Boundary geometry focuses on the surface or **shell** of objects, and ignores the **volume** and inner structure.

This is similar, for example, to making a photographic portrait of a person. We are mostly interested in capturing expression, gesture, skin texture, posture, eye color, and other details. In general, we are quite uninterested—as far as completing a successful portrait—in whatever is under the skin of this person: muscles, bones, and organs.

On the other hand, when we build three-dimensional computer models for the ultimate goal of fabricating them—with a computer-controlled milling machine or stereo lithography system—we are fundamentally interested in the inside of the object, its structural soundness, and whether the shapes we have included in our model can actually be fabricated efficiently (Figs. 15.1.1, 15.8.4, and 15.8.5.) For all these reasons when we model objects for CADAM projects we often use **constructive solid geometry** (CSG) techniques. These techniques are not concerned with how fast a three-dimensional model would render, how realistic it would look, or how efficiently it would animate. Instead, CSG techniques focus on whether our three-dimensional model meets structural criteria, whether it has the exact required dimensions, and whether it contains the specified amount of material.

Modeling Is Related to Rendering and Animation

The life of a three-dimensional computer-generated model rarely ends with the modeling process itself. Most three-dimensional models go on to be rendered, and many continue through the production process to animation. As you will read later in this book many creative and technical decisions made during the modeling process can simplify, complicate, or even paralyze the rendering or animation processes or both. Keep in mind that before you start to model, you should get as much information as possible about the plans regarding the rendering and animation of the geometry.

For example, a certain rendering technique, production deadline, or camera position may require that you cut in half the number of polygons used to define a section of the object, or the animation script might require that you group the objects a certain way. If you know about either of these conditions in advance, you will avoid the difficulty of taking apart a model that is finished in order to try to reduce the number of polygons or having to undo a complex five-

3.6.3 Sketching your ideas and trying different proportions and shape variations can help to refine your ideas before the beginning of modeling production. (Doll designs created by Carlos Grangel for the short *Alma*, written and directed by Rodrigo Blaas. © Rodrigo Blaas.)

level hierarchical structure with hundreds of objects in order to reestablish some basic links in a different way. Some game engines require the same geometry at different **levels of detail (LOD)**—this can sometimes be accomplished using **polygon reduction** software (Fig. 3.6.7).

In summary, keep the rendering and animation requirements of your project in mind during the modeling process. This will help to keep wasted time to a minimum.

Write Your Numbers Down

Writing your numbers down can often help both before you start modeling and throughout the modeling process. Initially it is important to write down (on a simple piece of paper, in the project spec sheet, or in your production journal) the numbers that describe general but important things such as: the general dimensions of your object, its position in three-dimensional space, and the boundaries of the active area or workspace. Writing down this type of information can be especially useful when you return to a project that was put on hold for a long time or when somebody who is not familiar with

3.6.4 The real-time model of *Spyro the Dragon*™ has 352 polygons and 230 vertices. The modeling started with a couple of polygonal geometric primitives and continued by pulling points. Individual polygons were added for features like the eyes, feet, and interstices between joints. (Spyro the Dragon™. Images courtesy of Universal Interactive Studios, Inc. and Insomniac Games, Inc.)

3.6.5 This *TMNT* character was modeled with polygons as proxies, and rendered as subdivision surfaces. (Teenage Mutant Ninja Turtles and TMNT are trademarks and copyrights of Mirage Studios, Inc. *TMNT* © 2007 Imagi Production Limited.)

the project has to take over because you have decided (or somebody has decided for you) to work on something else.

Save Your Work Often

Save your work often, every 15 minutes or so, and make periodic backups of your important data files. Some applications automatically save the file(s) that you are working on at regular intervals specified in advance. Take advantage of such features.

Check the Preferences File

Remember that both the three-dimensional modeling program that you are using as well as your computer's operating system keep their preferred, or default, settings in a **Preferences file**.

The contents of the Preferences file are important because they control directly and indirectly the result of many operations, functions, and tools. Some of the settings contained in a Preferences file may include, for example, the units used to specify the dimension of objects being modeled, or whether a tool for creating cubes will define by dragging from the center of the cube outwards or from one corner of the cube to the opposite corner. As you can see, some of these settings may affect the result of your three-dimensional modeling, rendering, and animation.

3.6.6 Computer visualization of a CADAM model built with Rhinoceros software. The stereo-lithographic model and the final bronze casting are shown in Figure 15.8.4. (© 1999 Bathsheba Grossman.)

3.6.7 The low resolution geometry model (top) might be suitable for scenes where the beetle is far away from the camera, but the high resolution version (bottom) might be better for close-up shots or for scenes where the beetle is the main character in the scene. (Polygon reduction and rendering by VSimplify software. © 1999 Virtue Ltd.)

In general, the last person who opened a file or who used the program or the computer system is capable of altering the files by changing the Preferences file. In some systems, Preferences files are attached to the three-dimensional computer program and in some cases to the model files themselves. Check your system for details.

Check Your Memory Requirements

Most of today's three-dimensional modeling software will allocate enough of the system's memory (RAM and/or virtual). This means that in most cases you do not have to be concerned about whether there will be enough space in the computer's memory for you to build your model. But sometimes, especially when complex three-dimensional models are being created in average computer systems, the issue of not having enough memory can become a problem. Most professional three-dimensional software today is comfortable with about 2 GB of RAM, but happier and faster with as much RAM as possible. In addition, having a fair amount of free space on the hard disk for swap memory is almost a necessity. Sometimes when the system does not automatically make sure that there is enough memory for you to keep building, the system will unexpectedly run out of memory and crash. Keep in mind that many modeling programs recover from errors gracefully (and will allow you to restore your data), but some do not.

(Next page) Detail of saxophone modeled with myriad techniques. (Courtesy of Toru Kosaka, STUDIO EggMan.)

Chapter 3

Key Terms

- Absolute values
- Axes
- Azimuthal coordinate system
- Boom
- Boundary modeling techniques
- Cartesian coordinate system
- Computer-aided design and manufacturing
- Concatenated transformations
- Constructive solid geometry
- Conversion utilities
- Cut-copy-paste
- Data file
- Dimensions
- Direct numerical description
- Dolly
- Drawing Exchange Format
- DXF
- Edge
- Export
- Facets
- File conversion
- File formats
- Foreign-to-native
- Front plane
- Geometric transformations
- Geometry files
- Geometry modeling techniques
- Global coordinate system
- Global transformations
- Hierarchy
- Import filters
- Left-handed coordinate system
- Levels of detail
- Line
- Local coordinate system
- Local transformations
- LOD
- Mathematical operations
- Modeling
- Native file format
- Navigation
- Nonproportional scaling
- Object coordinate system
- OBJ, .obj
- Objects
- Open Inventor
- Pan
- Perspective projection
- Pitch
- Planar surface
- Point
- Polygon, reduction
- Portable
- Preferences file
- Proportional scaling
- Rectangular coordinate system
- Relative values
- Right-handed coordinate system
- Roll
- Rotation
- Scaling
- Scene, Scene description language
- Shell
- Side plane
- Spaces
- Spherical coordinate system
- Structures
- Three-dimensional space
- Tilt
- Top plane
- Transformation matrix
- Translation
- Truck
- Universal file formats
- Vertex
- Volume
- Virtual Reality Modeling Language
- VRML
- Wireframe rendering
- Workspace
- World coordinate system
- World origin
- X3D
- Yaw
- Zoom

CHAPTER 4

Modeling Techniques

Summary

THE BASIC TECHNIQUES FOR MODELING three-dimensional objects with a computer system are covered in this chapter. The chapter starts with a short overview on curves, their use in the creation of surfaces, and the general differences between polygonal meshes and curved surfaces. Following that is a discussion of the basic geometric modeling tools available in most of today's systems. After that comes a survey of several derivative techniques including revolving, extrusion, and sweeping. Techniques for creating terrains and simple free-form objects are followed by a survey of utilities that are useful to modelers of all levels. An overview of modeling for real-time display concludes this chapter.

QUICK INDEX

4.1 Introduction

Just like traditional modeling techniques, the computer-based three-dimensional modeling process begins with an idea. Before the modeling process can start we try to visualize this idea of what we want to create, perhaps by creating some sketches or even detailed blueprints.

The conceptualization and design of the three-dimensional model usually constitutes the first stage in the simulation of a three-dimensional scene with a computer. From an artistic point of view this step is probably the most important one in the process because it is here where the basic characteristics of the scene are laid out: the shape, position, and size of the objects; the colors and textures; the lights; and the position of the camera. It is also at this stage where the basic ideas have to be analyzed and the best modeling methods chosen for each task.

I usually prepare the initial sketches that describe the three-dimensional objects and environments with traditional media, such as colored pencils or markers on paper. These sketches indicate the general characteristics of the objects such as size, relative position, color, and lighting effects. Once the sketches are finished, I analyze

(Top: Detail of *Monster Samurai*. © Sprite Entertainment, Inc.)

them and prepare a set of blueprints containing one or several detailed views of the object with dimensions (Fig. 3.6.3).

There are many ways of translating the visual information contained in the sketches into numerical information suited for computer manipulation. Most three-dimensional modeling software allows users to build the model interactively, providing the visual feedback that is so important in developing the shapes of objects and the layout of three-dimensional spaces.

When the modeling process is complete we usually end up with a file that contains a detailed description of the objects in our environment including information regarding their geometry, position, and hierarchy. Realistic images of the files can be obtained by rendering the file with some of the techniques presented in Chapters 6 through 9. There are many techniques for describing three-dimensional structures, and each produces different results and requires a different type of approach (Figs. 4.2.6 and 4.2.7).

4.2.1 The characters from this Xbox commercial are built with different types of lines and surfaces. The dialog box shows conversion options between different types of lines. (Image courtesy of Blur Studio.)

(ReBoot® and © 1997 Mainframe Entertainment, Inc. All rights reserved.)

4.2 Curved Lines

Lines are used to define the shape of the object and many of its surface characteristics. Lines are a fundamental component of all three-dimensional objects. For this reason it is important that we understand differences between types of lines, as well as their attributes and limitations. This section offers a brief characterization of some of the standard lines most commonly used in three-dimensional modeling. The names used here are as general as possible, but your software may use a slightly different name for a specific type of line or a line tool. The classification presented here is based on the practical characteristics of different types of lines, on their advantages and disadvantages, and also on their mathematical nomenclature.

One obvious difference between lines is that some are straight and some are curved. Straight lines define the shortest distance between two points, but curves are about subtlety of change and elegance of design. There are many differences between straight lines and curved lines: their mathematical description, their behavior as they are used to model, the type of three-dimensional structures they yield, and in most cases, their visual appearance. Some three-dimensional modeling computer programs are capable of converting curved lines to straight lines and vice-versa, but in many cases the results of these automatic conversions are sometimes visually surprising and might require considerable work before they "look good" and can be used (Figs. 4.2.1 and 6.4.1).

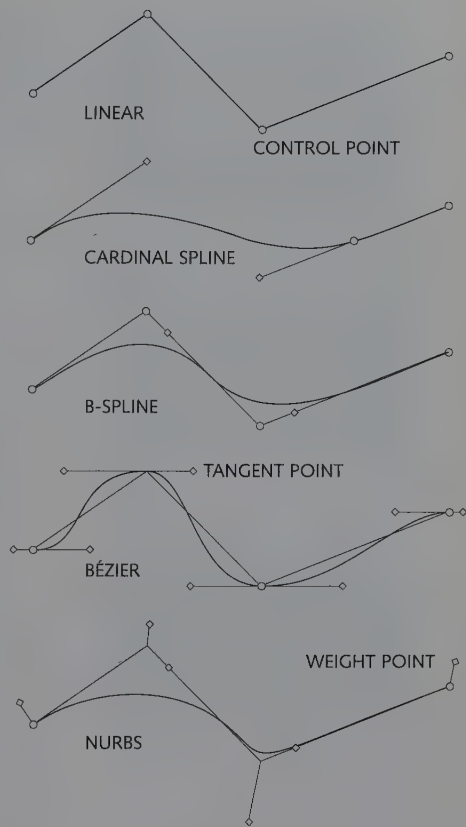
Straight lines—as their name implies—do not have any curvature. Straight lines are defined by two endpoints only, and may have a slope but no change in angularity. In other words, the slope of straight lines is constant while the slope of curves is variable. In three-dimensional modeling programs, straight lines are sometimes called **polygonal lines** because they can be used to build polygons and polygonal meshes. The three-dimensional modeling computer

programs that use exclusively straight lines are capable of building models only with polygonal meshes (and not with spline-based surfaces). Most three-dimensional modeling computer programs that use curves are capable of building models with both curved surfaces and polygonal meshes.

It is easy to turn a curve into a straight line by setting the change in angularity to nothing, but it is difficult to convert a straight line into a curve because straight lines just do not have a variable for change of angularity. For this reason, many three-dimensional modeling programs offer just a single curve line tool that draws curves of all kinds, including curves that look like straight lines.

Curved lines are usually defined by several points and deviate from a straight path without any sharp breaks in angularity. Curved lines are sometimes called curve segments and can be used to define curved surfaces and build meshes of curved surfaces.

Curves are also often called **splines** because they resemble the physical spline—a long narrow strip of wood or metal—used by a draftsperson or construction worker to shape or fit curves between various points. The spline, traditionally used in the design and construction of ships' hulls, is shaped by lead weights. By varying the number and position of weights the spline can be turned into a smooth curve that passes through the required points. Even though not all curves fall into the mathematical category of spline curves, some three-dimensional modeling programs use the term generically. From a strict mathematical point of view that generalization is inaccurate.



4.2.2 Five popular types of splines are illustrated here: a linear spline with control points, a cardinal spline, a B-spline, a Bézier curve with characteristic tangent points, and a NURBS, or nonuniform rational B-spline showing its characteristic weight points.

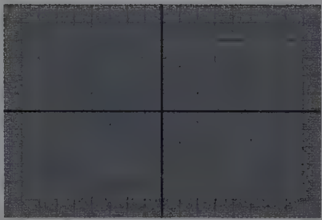
Five Popular Types of Curves

There are many types of curves, and they can be catalogued based on their mathematical and geometric characteristics. In this text, however, we shall limit our summary to five of the more popular types of splines used in mainstream three-dimensional modeling: linear splines, cardinal splines, b-splines, Bézier curves, and NURBS or nonuniform rational b-splines. Figure 4.2.2 illustrates these five types of curves.

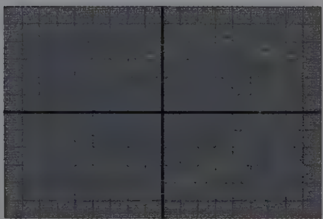
All splines are generated from a defining polygon. Because of this fact splines are called **controlled curves**. The structures that control the splines are invisible—they are displayed only while we shape the spline—but they contain important information that can be used to reshape the spline.

The controls found in splines of different types include the control line or control polygon or hull, the control points or control vertices, the tangent points, the knots, and the weights. Keep in mind that different software programs use different nomenclatures and slightly different implementations of the controls.

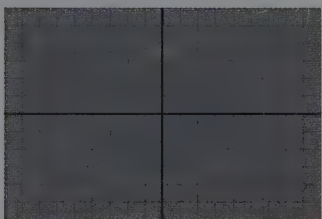
Each of the spline curves can be quickly characterized by the way in which it is controlled by the **control points** or **control vertices**. A **linear spline** looks like a series of straight lines connecting



SECOND DEGREE



THIRD DEGREE



FIFTH DEGREE

4.2.3 Graphs of NURBS curves of second, third, and fifth degrees. (© Alias|Wavefront.)

4.2.4 (Opposite page, top) This sequence illustrates how control and tangent points (T) affect the tension of a Bézier curve. In the middle drawing only the tangent points are moved. In the bottom drawing the control points A and C are moved closer to one another and the tangent points change accordingly to make the final curvature smoother.

the control points. A **cardinal spline** looks like a curve that passes through all of its control points. The **b-spline** looks like a curved line that rarely passes through the control points. A **Bézier curve** passes through all of its control points. A **NURBS,** or nonuniform rational b-spline, does not pass through its control points (Fig. 4.2.3).

Another easy way to characterize splines is to look at their controls other than the control line and the control points. Control points can control the **curvature** or **tension** of a curved line mainly by how close they are to one another and, in some cases, by how close they or their tangent points are to the curve (Fig. 4.2.4). The Bézier curve differs from the three splines mentioned here so far mainly because it has **tangent points** in addition to the control points. Tangent points are used to fine-tune the degree of curvature on a line without modifying the control points.

NURBS offer a high degree of local curve control by using weights and knots. These controls allow a portion of the spline to be modified without affecting other parts of the spline. One **weight** is attached to each control point, and they determine the distance between the control point and the apex of the curve. By default, all control vertices on a spline have the same weight factor, and that is called a **nonrational curve**. (B-splines, for example, are NURBS with equal weights.) When the values of the weights on the curve are modified then the curve is called a **rational curve**. Manipulating weights on a NURBS curve may improve the subtle shaping of a line, but it usually also slows down the rendering of the final model. Another disadvantage of working with different weight values is that many systems will ignore the data when model files are exchanged. In many cases, results similar to using different weight values can be obtained by placing two control points very close to each other.

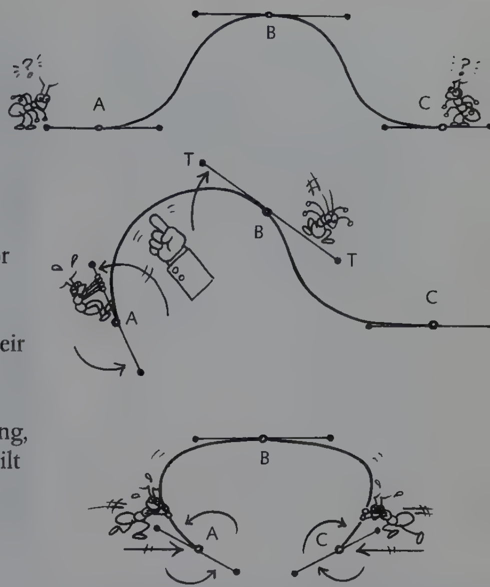
The **knots** on a NURBS determine the distribution and local density of points on a curve. The minimum number of knots required to form a curve segment is equal to the degree of a curve plus one plus the number of control points. The **degree of curve** refers to the high exponent in the mathematical formulas that generate curves. Each curve type (b-spline, Bézier, and NURBS) has a different mathematical formula, and each curve type may be created at different degrees (Fig. 4.2.5). The higher the degree of a curve the more computation is required to create it. Curves of the first degree correspond to linear segments, curves of the second degree correspond to **quadratic curves**, and curves of the third degree correspond to **cubic curves**. The higher the degree of a curve, the more control points and knots are required to form a curve segment.

4.3 Geometric Primitives

Virtually all three-dimensional modeling computer programs provide a collection of tools for creating simple shapes with a fixed structure known as **geometric primitives.** The number and type of geometric primitives varies from program to program, but the following list is a

representative selection: cubes, spheres, cylinders, cones, toruses, regular polyhedra, and two-dimensional polygons. Figures 4.3.1 and 4.4.1 include some of the most common geometric primitives. Geometric primitives are standard shapes that the modeling program can create and manipulate effortlessly and usually from a simple predefined mathematical description. Geometric primitives may be created as polygonal structures or as curved patches.

Geometric primitives can be used to represent simple shapes, or they can be used as the basis for more complex, composite three-dimensional shapes. In the former case, the shapes provided by the tool would require almost no modification except for changes to their position in space, size, and proportion in some cases. In the latter case, geometric primitives may be modified or used to build more complex objects with a variety of utility tools for trimming, attaching, and blending, among other things. Expressive characters can be built out of geometric primitives (Fig. 10.2.4).

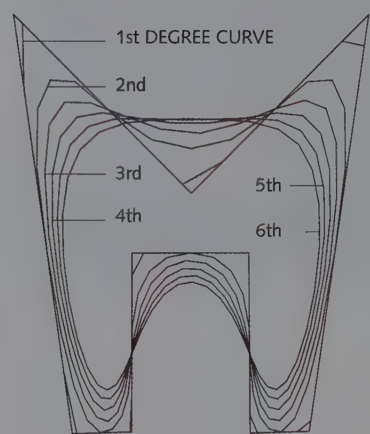


Cubes

Cubes are usually modeled as six-sided, closed, three-dimensional objects. Since all sides of a cube have the same length, usually the only variable required for modeling cubes is the length of a side. Sometimes a number of subdivisions can be specified along each of the three axes. Cubes are almost always created as polygonal structures.

Spheres

Spheres, like cubes, are also modeled as symmetric, closed, three-dimensional objects. In order to be defined, all spheres require a variable of **radius** or **diameter**, and they can be modeled as a polygonal structure or as a patch of curves. When modeled as polygonal structures, drawn with straight lines, a sphere's definition requires a number of divisions along the longitude (top to bottom) or latitude (around). These divisions resemble the parallels and meridians on a globe, and their number has a proportional effect on the geometric smoothness of the final shape. When modeled as curved patches, spheres require a type of spline to be specified in addition to the information listed above. Spheres are also very popular as the starting point for free-form modeling (Fig. 4.5.1).



4.2.5 In general, the higher the degree of a spline, the farther away it is from the controlling polygon (the outer shape with sharp angles).

Cylinders and Cones

Cylinders and **cones** are commonly defined as polygonal objects, and they may be shaped by the following variables: radius, height, number of longitudinal divisions, number of latitudinal divisions, and whether they are capped or not. The number of subdivisions used to build cylinders and cones defines the amount of modeling detail of the these objects. Objects with a small number of subdivisions can be rendered more quickly than objects with many subdivi-

4.2.6 The stylized characters and city in the movie based on Enki Bilal's *La femme piège* graphic novel are modeled with a combination of curved surfaces and geometric primitives. (Image courtesy of Duran.)

4.2.7 The main character in *Bingo* is modeled with a wide variety of techniques to define surfaces that must deform with both keyframe animation and cloth dynamics simulation. (© Alias|Wavefront, a division of Silicon Graphics Limited.)

sions. When planning to render objects with image maps it is best to model them with a large number of subdivisions, especially when squash and stretch animation is involved. Doing small rendering tests is essential to determine the optimum number of subdivisions. **Capping** determines whether the round sides of cones or cylinders are open or whether they are closed.

Toruses

A **torus** is a three-dimensional, closed shape that resembles a donut. A torus is like a cylinder that has been bent and stretched so that the two bases touch each other. The variables required to construct a torus are almost the same as those required for building a sphere, plus one additional variable, which is the interior radius. A full listing of modeling variables for a torus includes whether polygons or patches will be used, size of exterior radius, size of interior radius, number of latitudinal divisions, and number of longitudinal divisions. A torus is a geometric primitive but it can also be built, whole or sectioned, with radial sweeping techniques (Fig. 4.4.1).

Regular Polyhedra

Many three-dimensional objects belong to the category of **regular polyhedra**, or objects with multiple facets. A polyhedron (singular of polyhedra) refers to a three-dimensional object that is composed of polygons. Some of the most common regular polyhedra include the 4-sided **tetrahedron**, the 8-sided **octahedron**, the 12-sided **dodecahedron**, and the 20-sided **icosahedron**. Regular polyhedra are usually modeled as polygon meshes and can be built by specifying a radius and a number of facets required.

Two-Dimensional Shapes

An assortment of **two-dimensional shapes** can be used to generate three-dimensional shapes by using derivative techniques such as extrusion or sweeping. Two-dimensional shapes usually include arcs, circles, spirals, triangles, squares, and other polygons.

Circles are two-dimensional, closed contours and require a radius or diameter, a number of control points, and a type of spline. **Arcs** are two-dimensional, open contours and require the same information that circles do plus the starting point and the ending point, both specified in degrees. **Spirals** are also two-dimensional, open contours and require a starting and an ending radius, a starting and an ending angle, a number of control points, and a height. **Polygons** (including triangles and squares) are two-dimensional, open contours, are almost always built with polygonal or linear splines, and can be defined by a number of sides and radius.

4.4 Sweeping

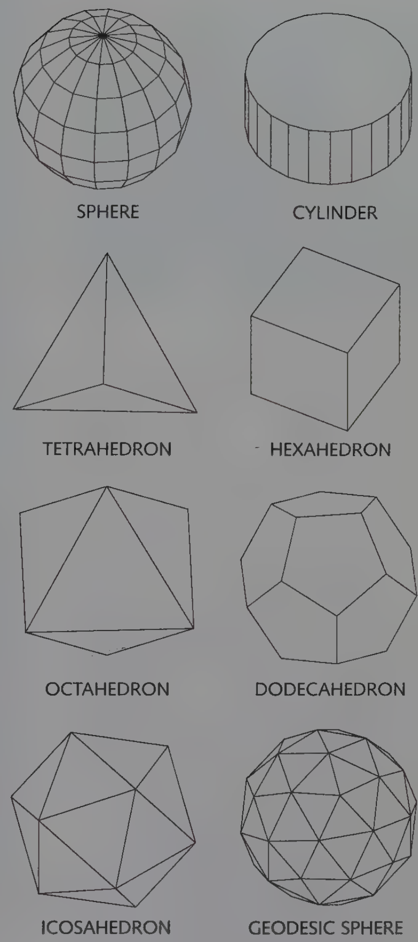
Sweeping is perhaps the most powerful derivative modeling technique, especially when you consider the complexity of the three-dimensional shapes created with it in relation to the simplicity of the input that is required for generating them.

The basic idea behind all sweeping modeling techniques consists of defining a two-dimensional outline that is swept along a predefined path. As the outline is swept it defines a shape in three-dimensional space. The resulting three-dimensional model depends largely on the complexity of the **seed outline** and the complexity of the path (Fig. 4.4.2). The three most popular sweeping techniques are extrusion, lathe or revolve, and free-form sweeps.

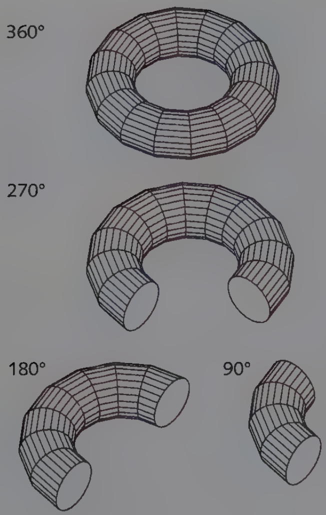
Simple Extrusion

In the conventional lingo of industrial design and manufacturing, **extrusion** stands for the process of shaping a material (such as plastic or metal) by forcing it with heat and pressure through a **die**. A die is a tool used for shaping or stamping different materials. The process of industrial extrusion is usually based on a stationary die just because of the limitations in handling both the hot materials and the heavy die. Meat grinders or pasta machines, for example, extrude the ground meat and the pasta through dies of different shapes.

Most three-dimensional modeling computer programs offer simple extrusion tools that—like their heavy industry counterparts—create three-dimensional shapes by starting with a two-dimensional outline and extruding or extending it along a **straight path** along one axis (Fig. 4.4.2). Simple extrusion happens along any one axis. The two-dimensional outlines to be extruded can be created with geometric primitive tools or exported from other programs in highly portable file formats such as EPS (Encapsulated PostScript).



4.3.1 A sphere, cylinder, and regular polyhedra including a tetrahedron (4-sided), a hexahedron (6-sided), an octahedron (8-sided), a dodecahedron (12-sided), an icosahedron (20-sided), and a geodesic sphere.



4.4.1 Toruses and other geometric primitive shapes can be created by sweeping a two-dimensional outline around an axis. Slices of geometric primitives are usually created more easily and faster this way than when geometric primitives are sliced with a trimming tool.

4.4.2 (Opposite page) The outline of the different floors were defined on the XY (top) plane, and the resulting outline was extruded along the Z axis. Details were added later using a variety of techniques including trimming and Boolean operations. The finished rendering was composited with an open shutter still photograph of the area where the building was scheduled to be built. (110 Bishop Gate. © Hayes Davidson.)

Extrusion is sometimes called **lofting** because the two-dimensional outlines are duplicated and moved a level up.

Free-Form Sweeping

Some programs also offer the ability to extrude objects along **curved paths** of any shape and along any axis or combination of axes. An extrusion that takes place along several axes is sometimes called a sweep, sometimes called an extrusion on a path, or a **free-form extrusion** (Fig. 4.4.3). The results of free-form extrusion that is scaled along the path or based on two paths are similar to those obtained with the skinning modeling technique described in Chapter 5.

Modeling by extrusion has been quite popular for centuries for creating meringue and ornaments on pastries, cookies, and cakes. The pastry extrusion tool, or die, moves with a sweeping motion along a decoration path. The motions of the pastry tool usually extend on a surface in a single continuous action, such as that of a broom or a brush, and in a wide curve or range.

Lathe or Surfaces of Revolution

One very popular sweeping variation is commonly referred to as a **lathe** or a **revolve**. This form of sweeping is so popular that it is almost always presented as a stand-alone tool, separate from the general-purpose sweeping tool. The surfaces created with this technique are usually called **surfaces of revolution**. The software-based lathe tool simulates a real lathe, which is a tool composed of a rotating base on which you place a cylinder of wood that is shaped by placing a steel blade on its surface as the base rotates around its vertical axis. A potter's wheel is used to perform an almost identical operation on a slab of clay (Fig. 3.2.1). The clay or wood are cut uniformly around the cylinder as a blade or sharp tool moves in and out following a predefined path. The software lathe sweeps a two-dimensional outline around one axis; the two-dimensional outline may be open or closed. A new three-dimensional shape emerges as the two-dimensional outline is swept along a circular or radial path; it usually remains perpendicular to the sweeping path as they are swept. The resulting three-dimensional object is defined by the areas enclosed within the revolved two-dimensional outline. Surfaces of revolution require an angle of rotation and a number of steps or facets. The number of subdivisions is usually determined by the number of points on the outline used to generate the shape.

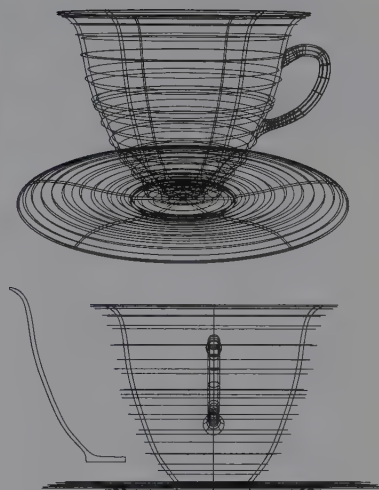
Surfaces of revolution that result from a 360-degree sweep are frequently closed, three-dimensional shapes. Sections—or slices—of three-dimensional shapes can also be created by sweeping less than 360 degrees. Two-dimensional outlines that do not touch the axis of sweeping will result in three-dimensional objects with holes (Fig. 4.4.4). In these cases, or when only a slice of a shape is created, the resulting shapes can be capped or uncapped.

4.4.3 (Above) Fluid shapes can be created by sweeping a free-form shape along multiple axes. These wolves have a unique primitive quality, different from both the cartoon and the realistic styles of modeling. (Pastilles Vichy. Agency: EURO RSCG BETC. Director: Pierre Coffin. Illustrator: Jean-Christophe Saurel. Images courtesy of Ex Machina.)

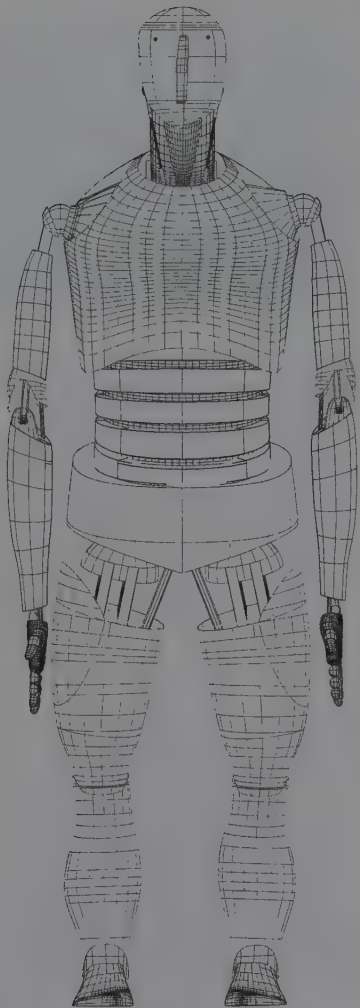
The lathe modeling technique can also be used to create simple geometric shapes such as the cylinder and the cone (Fig. 4.3.1). Using the lathe for this purpose offers the advantage of increased control and economy of steps when trying to model a special version or a variation of a geometric primitive.

4.5 Free-Form Objects

Some projects require the creation of **free-form three-dimensional objects**. Creating these types of models can be time-consuming because they must be sculpted from a mesh in a way that is very similar to sculpting or modeling a piece of soft clay. Simple free-form objects usually require a lot of point-picking, pulling-and-pushing, and overall "massaging" of the surface mesh (Fig. 4.5.1). The meshes may be planar, curved, or even based on subdivision surfaces. Planar or polygonal meshes, which are covered in this section, are particularly well-suited for gaming projects where real-time



4.4.4 This teacup was modeled by sweeping a two-dimensional outline 360 degrees around the Y axis, and adding a handle later. (Courtesy of Iris Benado.)



4.5.1 The *Straße der Spezialisten* robot was built with a combination of NURBS and polygonal primitives. See Figure 6.12.6 for the version rendered with ambient occlusion and a toon shader. (Image courtesy of Studio Soi.)

rendering almost requires that polygons are used. Medium resolution polygonal meshes look smooth at a distance, especially when mapped with detailed image maps, but the illusion of smoothness breaks down as the model gets closer to the camera. High resolution polygonal meshes like those obtained by scanning a three-dimensional maquette usually result in fairly large files, and they are sometimes used for animated feature films.

Free-form modeling—also called free-form deformation—is used when other modeling techniques are too rigid for building a specific scene or when using a combination of other tools would get the job done but would also require additional production time and a larger production budget. Free-form objects are also sometimes used because of the creative preferences of the individuals who design the look of the three-dimensional scene.

Free-form modeling techniques can also be used in conjunction with other techniques—especially those described in Chapter 5. The technique of free-form modeling has a couple of variations including direct point manipulation, deformation with lattices, and terrains.

Virtual Sculpting with Polygonal Meshes

The most common and easiest way to create three-dimensional free-form shapes usually starts with an existing three-dimensional structure that is to be "sculpted" and transformed into the desired free-form object. This **virtual sculpting** process is, in essence, quite similar to the process of modeling fresh clay with one's hands. The initial shape of the clay is fairly unimportant in terms of the desired final three-dimensional shape. But as our hands massage, push, pull, and rub the shapeless clay, it is slowly transformed into a meaningful structure. A simple primitive like a sphere can be the starting point, or we can start from a three-dimensional scan of a real-scale model. Some techniques for sculpting with free-form curved surfaces and for fitting curved surfaces to polygonal meshes are described in Chapter 5.

The **direct point manipulation** modeling process, as virtual sculpting is also known, starts by identifying the points—or control vertices—in the wireframe structure that can be displaced in three-dimensional space. Most three-dimensional modeling programs offer simple ways of picking a single point or a group of points. Direct point manipulation in the case of curved lines can be done directly to a point on the curve or to a control or a tangent point. Usually the selection is done by just clicking and dragging one or several points. Once the points have been selected they can be dragged in any direction. Some programs offer excellent tools for picking and manipulating single points. Some of the most useful direct point manipulation options include the ability to select several points that are not contiguous, or the ability to lock the position of points in some parts of the object while other points are being manipulated. Figures 4.5.1 through 4.5.7 illustrate the results of free-form modeling, and the different approaches to **shape styling**. Figure 4.5.1 illustrates a com-

bination of geometric and humanoid shapes, Fig. 4.5.2 is a sculptural cartoon, Fig 4.5.3 an organic soft and cute cartoon, Fig. 4.5.4 a delicate simplification of an idealized human body, Figs. 4.5.5 and 4.5.6 a mixture of realistic modeling with cartoony features—with and without deep wrinkles, and Fig. 4.5.7 a metaphorical rendition of a fantastic character.

4.5.2 NURBS curves were drawn along the surface of a simple initial volume. As the curves were lofted and sculpted, with special attention given to the loops of mouth and eyes, the initial reference volume was deleted. The curves were then lofted as polygons to better define the edges, and additional parts (eyes, nose, mouth, ears) were added and connected. Deformers such as lattice were used to refine the style and proportions. The hair is a polygonal mesh with sculpted detail. The rendered version can be seen in Figure 9.4.5. (Image courtesy of David Tousek, Bohemian Multimedia.)

Deformation with Lattices

Direct point manipulation can be a very efficient technique when only a few points need to be manipulated or when the user is really skilled at free-form sculpting. There is another free-form modeling technique that can be more appropriate for the task than direct point manipulation—especially in cases where a uniform global deformation is desired or when the user does not have the skill or the time to manipulate a large number of points one at a time. This technique is called **deformation with lattices**. (Lattices are sometimes called bounding boxes, not to be confused with the boundary boxes described later in this chapter.)

A lattice is a structure of points and lines that controls the points in the model. We can think of the lattice as a structure of grids that is connected to the points in the model with imaginary springs. Therefore, when the grids—or points on the grids—are moved, they drag the object's points with them (Fig. 4.5.8).

Every point on the lattice is connected to one or several of the model's points. The ability to control the deformation of the object by moving one or several grid points in the lattice depends directly on the number of lattice points. A small number of points on the lattice results in very rough or global distortion. Lattices with a large number of control points can be used to apply very subtle—or local—distortion on the object controlled by the lattice.

Simple Terrains and Functions

A great variety of techniques is available for creating **terrains** that simulate or recreate natural or imaginary landscape surfaces. A great

variety of other techniques use **mathematical functions** for distorting those terrains. The simplest technique for creating a terrain consists of using a flat two-dimensional plane with XY subdivisions. Obviously, the more subdivisions on a plane, the more detail will appear on the final terrain model. As mentioned earlier, the position of points on the plane can be modified by direct point manipulation or by lattice deformation. Either of these techniques would be appropriate if the shape was supposed to resemble a natural terrain. But if you are trying to create a more fantastic terrain, the basic plane could be deformed with a mathematical function. Figure 4.5.9 shows terrains created by distorting a terrain with different functions.

Another technique for creating terrains consists of building a three-dimensional mesh based on two-dimensional contours that define an imaginary or real landscape. This technique is very data-intensive but also very efficient for creating accurate models of terrains. Because of their topological detail, terrains created with this technique are rarely distorted with mathematical functions (Fig. 5.1.4). A simpler technique for quickly building terrains consists of applying a black-and-white image as a displacement map to a three-dimensional flat plane (Fig. 9.6.2).

4.5.3 The main character from the *Pocoyó* TV series is made with about 13,000 triangles. It was modeled with polygonal meshes in Softimage XSI, and subdivision surfaces were used to soften the shape during rendering with Arnold software. A typical Pocoyo scene has around 200,000 triangles, before subdivision. A few scenes in the second season went up to 600,000 triangles, due to an increase in characters and props simultaneously in a scene. (Zinkia™ and Pocoyo™ © 2005 Zinkia Entertainment SL.)

4.6 Basic Modeling Utilities

In addition to basic modeling tools, virtually all three-dimensional modeling programs offer a set of basic utilities meant to complement the modeling process. Among them we find such useful techniques as naming objects and getting information about them, duplicating, snapping to grid, mirroring, displaying as a bounding box, calculating volumes, and creating text.

Getting Information and Naming Objects

Objects and components of objects in some cases can be named so that we can identify them faster. Many programs will automatically name objects as we create them with names like Cube 1, Cube 2, Cube 3, or Node 1 of 5, Node 2 of 5, and so on. But in some cases where quick and unequivocal identification is required it is best to use unique names. For example, when one of 50 ellipsoids representing balloons is the target of a child's dart, naming it "target balloon" instead of "Ellipsoid 37" could be useful to quickly identify it during the explosion sequence (Fig. 4.6.1). Another useful feature for quickly assessing detailed information about a specific object—such as its position, dimensions, and orientation—is the Get Information feature that presents information about the active object in numerical form.

Locking

Objects can be locked in a specific position, orientation, size, or spatial range. **Locking** an object or an object's element that is not sup-

posed to move can help streamline the modeling process. In most cases, objects can be switched from the locked position to the unlocked position without losing or modifying any other attributes.

4.5.4 This girl in a red dress was modeled with modo software. Rendering was done with LightWave 3D using radiosity, subsurface scattering for the skin, and the Sasquatch hair shading plug-in. (© BTD STUDIO Co. Ltd./ Keica, Inc.)

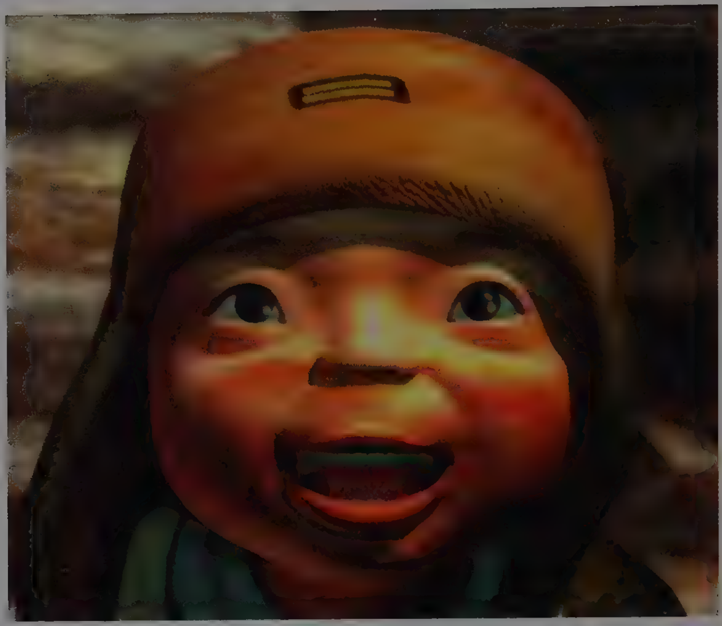
Setting a Face

Two-dimensional outlines drawn with free-form or curve tools are not really three-dimensional objects. When they are first drawn, two-dimensional outlines that are closed are just lines with a hole in the middle. Therefore, in order for two-dimensional outlines to be rendered properly it is necessary to turn them into planes. This process is called **setting a face** to an outline.

Setting the Center of Objects

By default, the centers of objects are automatically placed in the objects' geometric centers. Being able to interactively reposition this location is important because many modeling and animation operations are calculated based on the spatial position of the center of the object (Fig. 4.6.1). These operations include scaling, global and local

4.5.5 The *Birthday Boy* protagonist was modeled with soft shapes in a quasi-realistic but stylized way. Notice the skin's soft surface and subtle tinting as compared to the leather hat's textured surface. A long shot can be seen in Figure 7.2.9. (Image courtesy of AFTRS and Sejong Park.)



4.5.6 This character from *Ark* has escaped to the sea after an unknown virus almost wiped out the entire human race. His deep facial wrinkles contribute to express the stress and distress that he faces. Notice the rim light to accentuate the dramatic profile. (© Copyright by Platige Image. All rights reserved.)

rotations, linking, and simulations of motion dynamics related to the center of gravity.

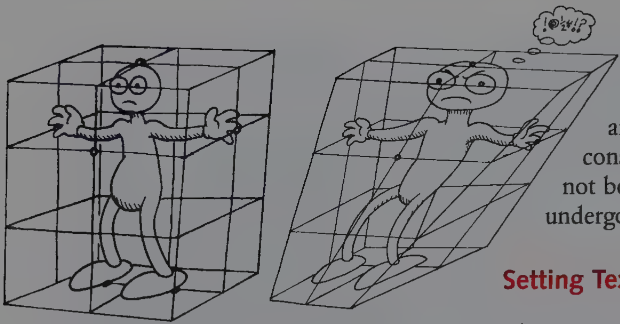
Duplicating and Instancing

Models can be easily duplicated without having to build them from scratch. **Duplicating** creates a single independent copy of the selected model or group of selected models. The copy can be created in the same location as the original or in a new position defined by an XYZ offset value. The duplicating utility can also create multiple copies of an object. The values needed to create multiple copies of an object

typically include the number of copies, as well as the XYZ values for translation, rotation, and scaling (Figs. 4.6.2 and 4.6.4). Creating copies of an object creates more three-dimensional elements in the scene, increases the file size, and demands more computing time.

Instancing is an alternative to duplicating that is available in many systems. **Instancing**—also called **cloning** in some systems—creates multiples of an original object by using its numerical description and cloning it elsewhere in the scene. The multiples created with instancing are like "living clones" that continue to be related at all times to the original object. If the original changes shape or is scaled, its dependent instances are also transformed. Because

4.5.7 Most of the branches and twigs in this character from the *Mother Nature* series were created with Maya's Paint Effects. All geometries are in the same scene but rendered in separate passes, including color, shadows, and specularity. Retouching and compositing was done with Photoshop over a low-resolution digital photograph. See a related image in Figure 5.5.10. (Image courtesy of Meats Meier.)



instances of a model do not increase the size of a file they are convenient for creating large armies of objects that look alike and that display a consistent group behavior. Instancing, however, may not be appropriate if a project requires each multiple to undergo a different shape transformation.

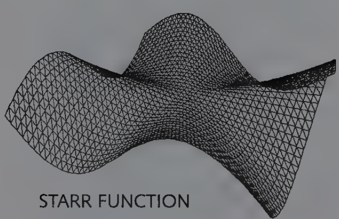
4.5.8 These drawings illustrate the effect of lattices on the shape of a three-dimensional object. Every time the lattice is moved the model is deformed because each of the grid points on the lattice is connected with imaginary springs to the object's points.

Setting Text

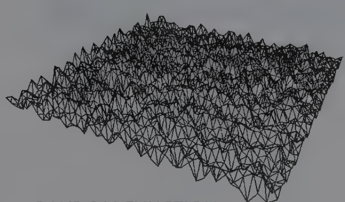
Text tools are capable of automatically producing two-dimensional outlines or three-dimensional objects extracted from the two-dimensional outlines of fonts (or typefaces) installed in the computer system. The sophistication and variety of two-dimensional text outlines varies greatly from software to software, and so do all the additional features associated with letterforms, such as letterspacing, kerning capabilities, and point-editing features.

Most of today's three-dimensional modeling programs extract the text outlines from spline-based descriptions (often in the PostScript language) resident in the system software. Some three-dimensional programs extract this outline information from a font database—sometimes in curve format, other times in polygonal format—that is provided with the three-dimensional program itself. The shapes of the letterforms are usually smooth and detailed when the outline information is brought in as a series of curves. However, when the outline information is brought in as a series of polygonal lines, the resulting shape may be jagged and unrefined, especially in the portions of the outline with the most curvature.

Any character that can be typed from the keyboard will show up in the three-dimensional environment as a two-dimensional outline, ready to be extruded and bevelled (Figs. 5.4.1 and 5.4.2).



STARR FUNCTION



RANDOM FUNCTION

4.5.9 A plane terrain with a resolution of 40 × 40 units is deformed with different mathematical functions: a Starr function, a random function, and on page 131, versions of the Julia and Mandelbrot fractal functions.

Mirroring

Mirroring a three-dimensional model is a useful technique when building an object composed of two identical (or almost identical) halves. The **mirroring** technique is implemented in a number of ways by different software, each with particular requirements and subtle functional differences. Three-dimensional objects can be repositioned in space with the mirroring technique. In such cases the object is entirely moved to where its mirror image would be. In general, however, mirroring works by copying an object, placing the copy in the same location as the original, and finally repositioning the copy. This way, the object remains in its original position, and its copy is placed where the mirror image of the original object would be.

Mirroring works by either providing a scaling value of –1 along the axis on which the mirroring is to take place, specifying a two-dimensional plane (XY, XZ, or YZ) on which the object is to be mirrored, or by establishing an axis of reflection by clicking a line perpendicular to the object to be reflected (one end of the line represents

the base point of reflection, the other end represents the beginning of the axis reflection). Mirroring is illustrated in Figures 4.6.2 and 4.6.3.

Snapping to the Grid

By forcing the object's or its components' points, for example, to **snap to a grid**, three-dimensional modeling programs can help to simplify the construction of regular shapes or precise details within larger shapes. Grids can usually be defined by the user. This includes the size of the grid unit, whether the points snap to the grid, whether the object's center or edges snap to the grid when the object is moved, and whether the snap-to-grid function is applied to all objects in the scene or only to some.

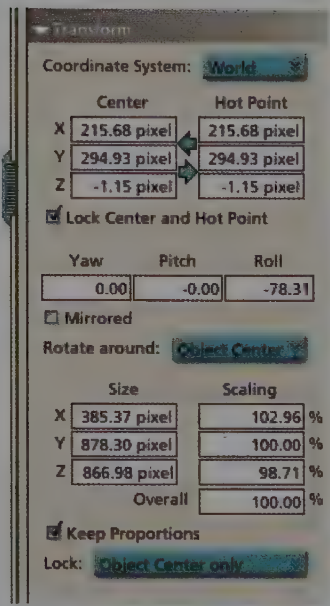
Volume Calculation

The calculation of volumes and unfolding of planes are two specialized modeling techniques that can be useful when designing three-dimensional models that eventually get fabricated out of real materials. Volume calculation tools allow users to find out the total volume and area of the inside, outside, or parts of any three-dimensional object. Knowing the exact volume of liquid that can be contained in a new bottle design can be very important to an industrial designer. Likewise, an engineer in charge of supervising the actual production of the bottle needs to know the volume of glass needed to fabricate the bottle. Some of the volume calculation tools can also be used to extract data related to the object's mass or its center of gravity and inertia. This information can be used later in the animation of models using motion dynamics animation software.

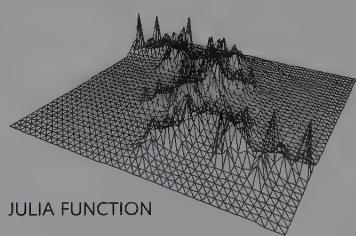
Being able to unfold the plans that bound a three-dimensional object can be quite useful when it is necessary to fabricate either a cardboard scale model or prototype of the three-dimensional object or the final object itself in more durable materials, such as plastic or sheet metal. Figure 15.8.1 shows a three-dimensional object that was built with a variety of modeling techniques and then unfolded into two-dimensional patterns.

Bounding Box

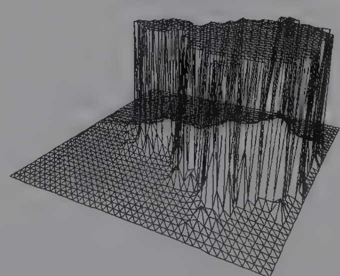
When modeling a scene with multiple complex objects many computer systems may slow down because of the huge number of calculations needed to redraw the image of the models on the screen. Using bounding boxes to represent objects is a convenient technique for speeding up their display. **Bounding boxes** are usually rectangular, and they are defined by the points most distant from the center of the model. Bounding boxes can also be used to define the **collision detection volumes** in a videogame (Fig. 4.6.4) or a dynamics simulation (Fig. 13.3.1). Bounding boxes are not to be confused with lattices, which can be used to distort the three-dimensional objects



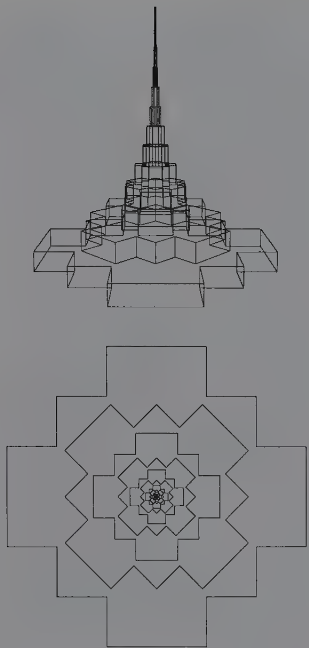
4.6.1 This dialog box provides quick numerical information about the position, orientation, size, and ranges of motion of an object. (© Eovia Corp.)



JULIA FUNCTION



MANDELBROT FUNCTION



4.6.2 This object was created by duplicating the original shape at the bottom 12 times. Each instance of the original shape was mirrored by rotating 45 degrees and translating one unit on the Z axis.

4.6.3 (Opposite page) Some components of these objects were created by duplicating and mirroring. Notice the shallow depth of field that was used for the camera in the final rendering. (Courtesy of Toru Kosaka, Studio EggMan.)

contained within them. Bounding boxes and the lattices used to deform free-form objects look similar but behave differently.

Making objects invisible or **ghosting** them are two options similar to the bounding box. Making objects invisible removes them from the display but not from the information contained in the file. Objects made invisible with this method are usually not displayed regardless of the rendering method until they are made visible again—usually by just clicking a choice in a checkbox. The ghosting option offered by some programs is similar to the bounding box. In some implementations of the ghosted display the model is represented with dotted lines, and the display of the ghosted model is only updated when the mouse button is released at the end of an interactive manipulation. In other implementations of ghosting it does not speed the display of images on the screen but instead facilitates working with complex models by making ghosted portions unselectable.

4.7 Real-Time Polygonal Models

The ability to render polygonal models in real time is almost as old as three-dimensional computer animation, but today's systems offer more realistic rendering and more complex models than what was possible a decade ago. Gaming is the area that takes the most advantage of real time rendering, whether it takes place on game platforms or computers. Interactive online websites with three-dimensional navigation and functionality also take advantage of polygonal models rendered in real time. Both gaming and online websites require an approach to modeling for real time that emphasizes efficiency and portability.

Improved hardware rendering, as explained in Chapter 6, has fueled the feasibility and popularity of ever more complex real-time polygonal models. Powerful graphics cards and GPUs, faster clock speeds, and increased memory continue to raise the bar for real-time polygonal models. But there are always limits to what a powerful graphics system can do, and for that reason it is important to **optimize polygonal models** for real-time rendering. Designing and building polygonal models and their textures for real-time rendering is about compromise and optimization (Fig. 4.7.6). There is always a compromise between visual detail and speed in a real-time environment. Real-time models may range from a few hundred polygons to thousands of polygons. ***Spyro the Dragon***, for example, has 352 polygons (Figs. 3.6.4 and 11.1.1) while ***Dawn*** (Fig. 6.10.6) is built with 203,741 triangles. A "heavy" model with too much geometrical complexity could slow down the real-time rendering and the frame rate would drop, lessening the illusion of natural motion. Same would be true for a model with image maps too large for the available memory, graphics card, or GPU in question. Keep in mind that the geometry and size of image maps that might be optimal for one environment might be overkill or not enough for another. This is a common challenge when "porting" or adapting a computer game from one platform to another, from platform to PC or vice-versa: models and image maps

sometimes require significant revision and optimization. The issue of heavy geometry and maps, however, becomes less of a problem as the power of hardware continues to increase.

Most computer or platform games use their own custom **rendering engine** to render polygonal models in real-time (Fig. 4.7.1). The engine is fed with highly optimized polygonal and image map information. Because different game engines use a variety of different rendering and animation techniques the polygonal model and image maps formats may vary between engines (Figs. 3.6.4, 4.7.5, 7.4.6, and 11.5.4). For this reason it is always a good idea to keep a lowest-common-denominator model that can be adapted and converted for different game platforms and engines.

Most online websites that use real-time polygonal models require a **player** or plug-in. Most multiplayer gaming or community websites use their own formats and engines (Figs. 4.7.2–4.7.4, and 4.7.8). There are also a few off-the-shelf players for real-time three-dimensional rendering, available as Web downloads. Some of these off-the-shelf formats and players include Shockwave 3D (Fig. 4.7.8), Viewpoint, Cult 3D, and MPEG-4. The later file format is used primarily for encoding moving images, but it also provides tools for encoding three-dimensional polygonal meshes that can be used for low resolution real-time character facial and body animation. MPEG-4 files may include compressed geometry, connectivity, shading normals, color, and texture coordinates, as well as visual lip configurations, called **visemes**, that are equivalent to speech phonemes.

In addition to using powerful graphics cards or simplifying the

4.6.4 The collision detection volumes in blue define the areas of vulnerability of a soldier in the game *Medal of Honor*, and trigger hit reactions. The larger boxes are used for the easy playing mode while the smallest boxes offer a greater degree of difficulty. (© 1999 Electronic Arts Inc. All rights reserved.)

4.7.1 *One* is a fighting game built for Nokia mobile phones equipped with N-Gage software. The fighting sequences were originally motion-captured, and are played by a custom engine with about 2,600 polygons per scene. Two players can compete in Bluetooth mode. (© 2008 Nokia.)

4.7.2 *Fishook* was built and optimized for real-time display in the 3D virtual world *Second Life*. The fashionable dress is made of hundreds of floating hooks. Impossible to build in our physical reality, it exemplifies the creativity of online communities users. (© Irena Mandic Morris/Eshi Otawara, 2008.)

geometry and the image maps, there are a few tricks and shortcuts that can be used to accelerate real-time rendering. A few of the most common are levels of detail, billboards, and Flash animations. Levels of detail are useful in situations where the same three-dimensional character or prop appears very close and very far from the camera. Levels of detail, also called LODs, consist of having the same model at different resolutions that can be seamlessly loaded as the object or character moves closer or away from the camera (Fig. 3.6.7). Image maps with different levels of detail can also be used to highlight details such as the tattoo in Figure 4.7.6. Billboards are a few polygons that define a flat surface used to project image maps. Billboards have the same function that painted backgrounds have on a theatre stage or a movie set (Figs. 4.7.9 and 4.7.10). (See Chapter 9 for more information on billboards for visual effects.) Using Flash MX files to deliver three-dimensional animations online is another common shortcut. In this case the three-dimensional animation is rendered and saved as a two-dimensional SWF file or Flash movie. The advantages of this approach, or "cheat," are that Flash files are generally compact and download fast, and also that the Flash player is ubiquitous worldwide. Three-dimensional animations saved in the SWF format are usually contained in small windows, 200 × 200 pixels for example, that can be imported into a Flash movie and easily played within larger scenes.

Many 3D computer animation applications and some game engines have adopted **COLLADA** as an asset exchange format. Google Earth, for example, has adopted it as its native format (Fig. 4.7.7). Originally developed for Sony's PlayStation 3, COLLADA became an open standard. COLLADA stands for Collaborative Design Activity, and its XML files use the .dae (digital asset exchange) extension. Some versions of the format are capable of

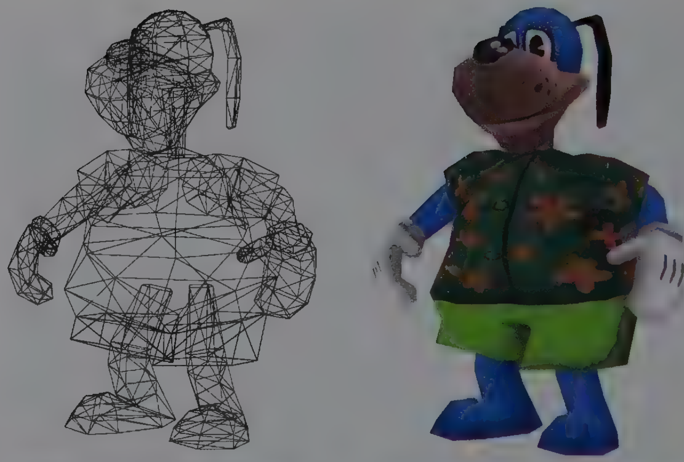
4.7.3 The multiplayer game *EverQuest* has characters ranging between 500 and 3,000 polygons, a minimum of two 256 x 256 textures, and bump maps. (EverQuest® courtesy Sony Online Entertainment Inc. EverQuest is a registered trademark of Sony Computer Entertainment America Inc. © 1999–2002 SCEA Inc. SOE and the SOE logo are registered trademarks of Sony Online Entertainment Inc. All rights reserved.)

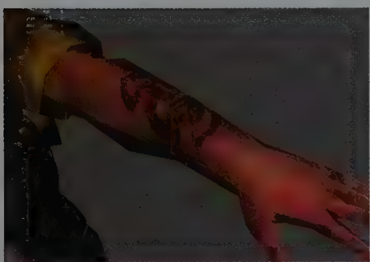
4.7.4 *URU, Ages Beyond Myst* is the multiplayer online version of *Myst*. The high resolution avatar is roughly 5,000 polygons, but level of detail (LOD) techniques are used to lower its resolution for long shots. The environment is several thousand polygons, but the total may surpass 15,000 polygons when the degree of tessellation of the water increases due to ripple and flow. (© 2002 Cyan Worlds, Inc. All rights reserved.)

storing material properties and physical attributes of a 3D model.

Computer and platform games are the most pervasive applications of real-time rendering, but certainly not the only one. Fine artists are increasingly creating interactive worlds and virtual reality installations using off-the-shelf players or repurposed game engines; the latter approach is called **game moding**. Figure 12.7.3 shows a virtual reality installation that requires users to wear a VR helmet. Figure 4.7.8 shows ***Purbeck Light Years***, a real-time computer animation and immersive virtual reality based on paintings and drawings. The land mesh in this installation was created from a satellite image of Dorset, England, with the grayscale values used as a displacement

4.7.5 Sample character in wireframe and shaded versions from Disney's *Toontown* online multiplayer game. Model is within the 1,000 polygon average, including a 392-polygon head, 476-polygon torso, and 158-polygon legs. (© Disney Enterprises, Inc.)





4.7.6 The topology of this polygonal mesh has been designed to optimize the folding of the geometry and image maps at the knees and elbows. The high resolution map of a tattoo was placed on a dozen polygons using UV coordinates. (© 1999 Electronic Arts Inc. All rights reserved.)

map on a 8,192-polygon terrain. The ground and sky are covered with bitmap textures. To simulate day or nighttime the ground and sky textures blend gradually into other textures. The skies are chosen at random to represent the unpredictable nature of real weather, and during the daytime there is a 30% chance of rain. The textures for the rectangular vertical planes, or billboards, are from drawings and paintings of Corfe Castle and an alpha transparency channel is used to display their irregular shapes. A flocking system governs the movement of birds, a theatrical element in the project. A polar coordinate system (Fig. 3.4.8) is used to keep the castle in the middle of the scene at all times.

4.7.7 Online 3D model for Google Earth of the Tower of Hercules, the oldest Roman lighthouse in the world still functioning as such, overlooking the North Atlantic coast of Spain. Modeled with Google SketchUp software. The model has 1,498 polygons including the terrain; 767 for the tower and base, 731 for the hill. The textures range in size from 19 x 183 to 800 x 500 pixels in JPG and PNG file formats. The model is available in Google's 3D Gallery in .skp, .kmz, and COLLADA file formats. (© 2008 Anxo Miján.)

4.7.8 In addition to the 8,192-polygon terrain in *Purbeck Light Years* there are 128 models in this Shockwave 3D world, each consisting of 2 polygons for a world total of 8,448 triangles. A higher resolution model, 131,072 polygons, was used to capture these images. The castle image maps are 32-bit color and 1024 × 512 pixels, and the maps for the planes, ground and sky are 512 × 512 pixels. (© 2003 Jeremy Gardiner. Programming by Anthony Head, research by Veronica Falçao.)

4.7.9 First-level playground in the *Toontown* multiplayer game, with two avatars that players can create interactively from a catalog of parts (torso, legs, head, muzzle) and clothes. (© Disney Enterprises.)

4.7.10 Wireframe and shaded versions of a corner in *Toontown*, where real-time environments typically range between 2,000 and 3,000 polygons excluding characters. The Panda3D proprietary engine can display geometry in real time using either OpenGL or DirectX. (© Disney Enterprises, Inc.)

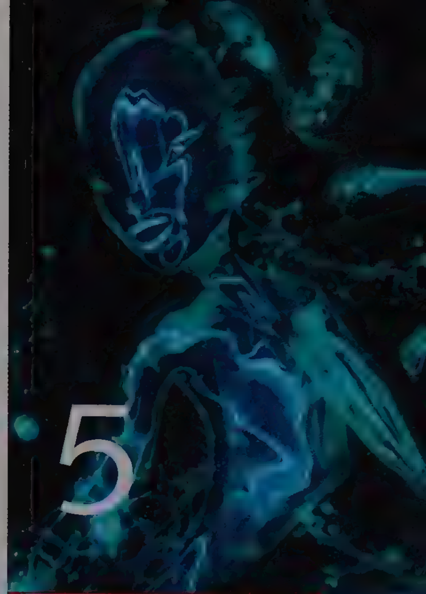
CHAPTER 4

Key Terms

- Arcs
- B-spline
- Bézier curve
- Bounding boxes
- Capping
- Cardinal spline
- Children
- Circles
- Cloning
- COLLADA
- Collision detection volumes
- Cones
- Control points
- Control vertices
- Controlled curves
- Cubes
- Cubic curves
- Curvature
- Curved lines, paths
- Cylinders
- Deformation with lattices
- Degree of curve
- Diameter
- Die
- Direct point manipulation
- Dodecahedron
- Duplicating
- Extrusion
- Free-form extrusion
- Free-form modeling
- Free-form three-dimensional objects
- Game moding
- Geometric primitives
- Ghosting
- Icosahedron
- Instancing
- Knots
- Lathe
- Linear spline
- Locking
- Lofting
- Mathematical functions
- Mirroring
- Nonrational curve
- Nonuniform rational b-splines
- NURBS
- Octahedron
- Optimize polygonal models
- Parents
- Player, plug-in
- Polygonal lines
- Polygons
- Quadratic curves
- Radius
- Rational curve
- Regular polyhedra
- Rendering engine
- Revolve
- Seed outline
- Setting a face
- Shape styling
- Snap to grid
- Spheres
- Spirals
- Splines
- Straight path
- Surfaces of revolution
- Sweeping
- Tangent points
- Tension
- Terrains
- Tetrahedron
- Torus
- Two-dimensional shapes
- Virtual sculpting
- Visemes
- Weight

CHAPTER 5

Advanced Modeling and Rigging Techniques



QUICK INDEX

(Top: *Frequency Morphogenesis*. © 2007 Filmakademie Baden-Württemberg, Onni Pohl, and Sebastian Schmidt.)

Summary

THE ADVANCED MODELING TECHNIQUES used for building three-dimensional objects and environments are covered in this chapter. These techniques include complex curved surfaces and blobby surfaces, logical operators and trimmed surfaces, a variety of utilities like surface blending, procedural description used mostly to model natural phenomena, and image-based modeling. An overview of animation rigging techniques concludes the chapter.

5.1 Free-Form Curved Surfaces

Free-form curved surfaces have been a popular modeling technique since the mid-1980s. Their smoothness and curvature are easy to manipulate with the mesh of control points associated with all curves. Unlike polygonal meshes that allow the free addition or deletion of single polygons, adding local detail to curved surfaces can sometimes be a task with multiple steps because curved surfaces must start with a checkerboard pattern where only entire rows or columns can be added or deleted. In the absence of great local modeling control, curved patches are commonly stitched to one another to create complex shapes. In the case of a hand modeled with curved surfaces, each finger, the thumb, and the palm would be modeled as separate patches and then stitched together.

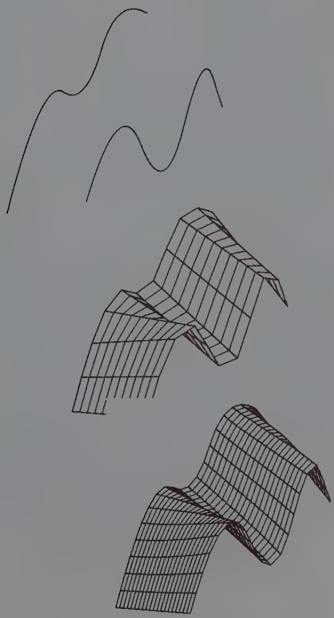
Free-form curved surfaces allow a great degree of surface control. They are mathematically defined and are also called **parametric curved surfaces** because each coordinate is a function of one or more parameters, such as a control hull, control points, tangent points, knots, and weights. **Free-form curved surfaces** and complex free-form surfaces are the generic names given to bicubic surface patches, Bézier surfaces, b-spline surfaces, and skinned surfaces. Each of these types of surfaces is based on different types of curves.

As explained in Chapter 3, curved lines are defined by several points and deviate from a straight path without any sharp breaks in

angularity. Curves are used to define free-form curved surfaces and to build meshes of curved surfaces. There are many types of curves, and the most popular types include linear splines, cardinal splines, b-splines, Bézier curves, and NURBS (Fig. 4.2.2).

Each of the curved surfaces can be characterized by the way in which its curvature or tension is controlled by the control points. Curved surfaces defined by Bézier curves, for example, pass through all of its control points, and their degree of curvature is fine-tuned with tangent points. Curved surfaces created with b-splines rarely pass through the control points, and those created with NURBS do not pass through the control points, but rely on weights and knots for increased local curve control. (Read Chapter 4 for more information on curved lines.)

Cubic splines make use of curve-fitting techniques and pass through all of the specified points. They also use parabolic blending and require numerical specification of both direction and magnitude of the tangent deviations. In the case of Bézier curves, the shape and order of the curve can be varied by the use of parameters. The first and last points are used to define the curve, which is defined by an open polygon. If any points are moved, then the entire curve is altered because it is an average of all its vertices. B-spline curves offer more local control than Bézier curves.

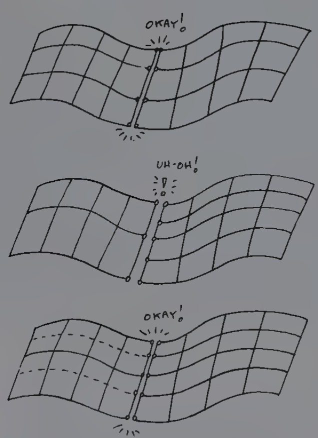


5.1.1 Two curves were used to create patches of different resolutions.

Curved Patches

A curved patch is a small curved area that can be created from two or four curves. When a patch is created from two curves, they are positioned opposite to one another (Fig. 5.1.1). Four curves can also be used to define a rectangular area. Complex free-form surfaces are created by **merging** two or more **curved patches**. When curved patches that have the same number of rows and columns are merged, the results are fairly predictable. But merging patches with different numbers of rows and/or columns requires the use of interpolation techniques that modify one of the two patches being merged (Fig. 5.1.2). This merging can also be controlled with great detail by specifying manually which points in the first patch merge into which points in the second patch. Having patches that merge into each other seamlessly is important to create a crisp rendering of the new patch, especially when the surface contains an image map on it.

Merging patches is one of the most powerful modeling techniques that can yield detailed models with subtle shapes (Fig. 5.1.3) However, the large number of points or vertices in a model created with patches is often a concern. This can be addressed by either making sure that the patches throughout the model have a small number of subdivisions, or by using a modeling utility that purges some of the points according to criteria such as the angle or distance to other points. Curved patches can be further manipulated and refined by using some of the utilities described both in Chapter 4 and in this chapter (Fig. 4.5.2).



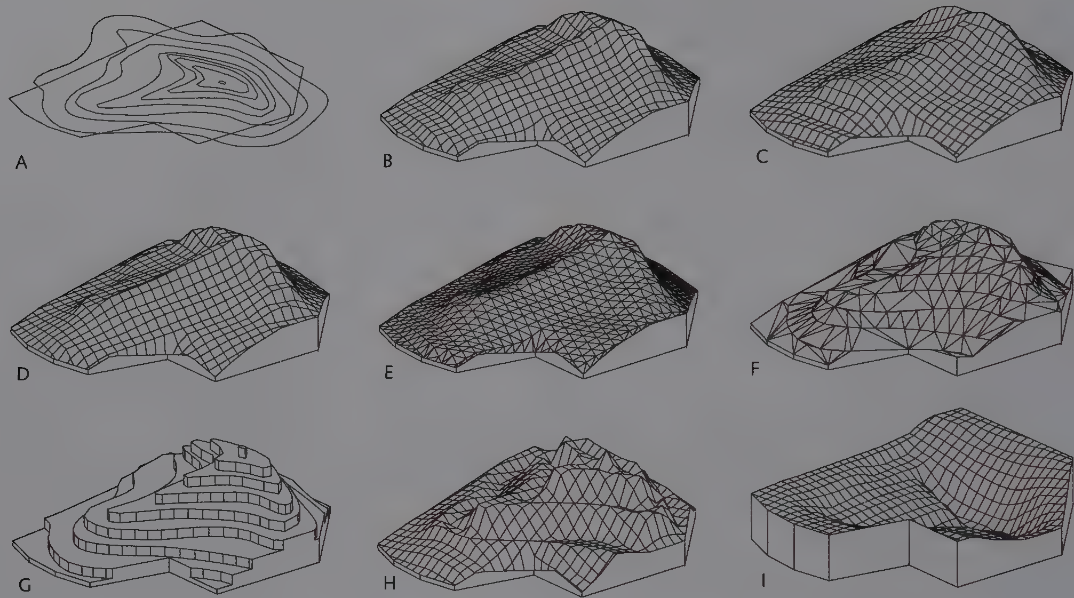
5.1.2 Merging two patches usually requires matching their number of columns and/or rows.

Skinning

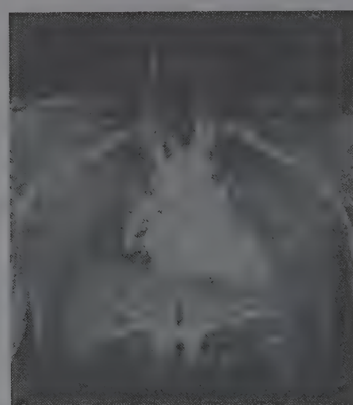
The technique of **skinning** creates three-dimensional objects by connecting a sequence of two-dimensional **cross sections**, also called **slices** or **outlines**, with curves. This modeling technique is not to be confused with the more general idea of creating a "skin" or deformable surface around an animation rig or skeleton. The final surface that is animated may be modeled with any number of techniques, including skinning, that eventually ends up as a finely-tuned mesh of polygons.

Skinning is somewhat similar to free-form extrusion because the modeling follows a path, but it is different because the shape of the cross sections being skinned is predefined. Most skinning functions usually require that the two-dimensional outlines are closed and that they all have the same number of points. Some skinning functions, however, can skin outlines with different numbers of points by creating points as needed based on guessing what the best skinning would be. Topographic data that includes altitude information is commonly used to generate terrain models using the skinning technique (Fig. 5.1.4). The technique of skinning is also called **cross section extrusion** or **serial section reconstruction** because entire objects can be reconstructed from cross sections or slices that can be obtained with a variety of methods. Skinning is particularly useful to create models of humans because these are easily described as series

5.1.3 This model of a famous actress was created by making a physical bust, scanning it, converting it to a NURBS surface, and then pulling points with Softimage, the software that was also used for animating the model. The single surface has openings at the mouth, the ears are separate surfaces, and the eyelashes were created with bump maps. (© 1999 Virtual Celebrity/Marlene Inc.)



5.1.4 (Above) Different three-dimensional objects can be created by connecting, or skinning, two-dimensional outlines (A) in a variety of ways. A mesh based on both XY directions (B) looks slightly different than meshes weighted along the X or the Y axis (C and D). The outlines in (E) and (F) are connected by two triangular meshes of different resolutions and by simple steps in (G). The altitude of every other outline has been shifted up or down in (H), and in (I) the altitude of the outlines has been inverted.



5.1.5 A three-dimensional scanner can quickly gather precise modeling data. (Courtesy of Cyberware, Monterey, CA.)

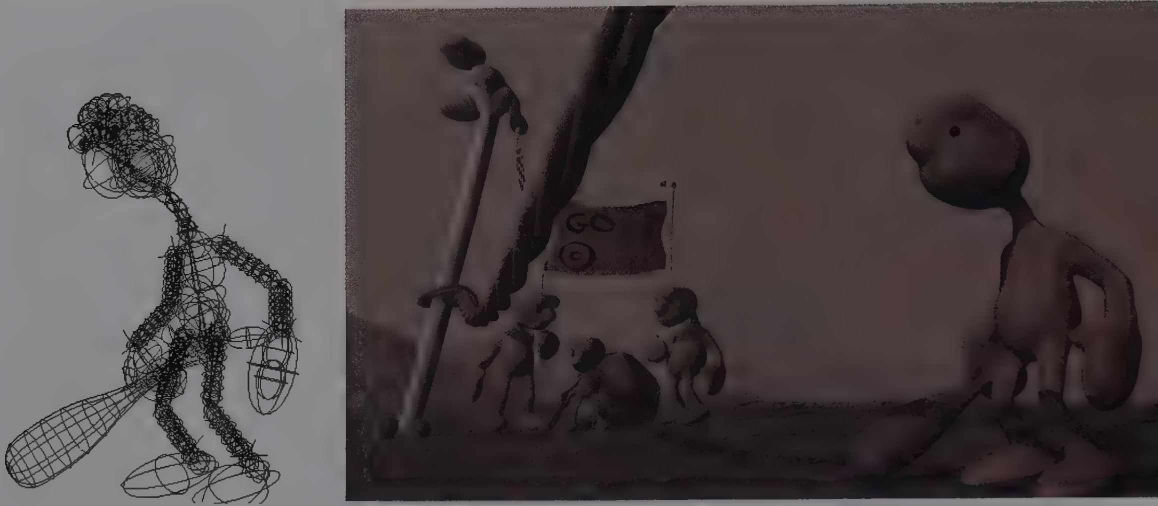
5.1.6 (Far right) Magnetic resonance (MR) scans were used to recreate the organs and structures inside a human thorax. Opacity settings were used to emphasize the bright flowing blood. (Courtesy of Irwin Sobel, HPLABS.)

of two-dimensional contours. The contours that define a skinned object can easily be traced manually on a digitizing tablet. This technique is especially useful for obtaining three-dimensional skinned objects that have the irregularities typical of handmade objects. The number of digitized cross sections needed to describe an object is in direct proportion to the complexity and detail desired in the final object. The sections on the X and Y planes represent a sample in the horizontal and vertical levels of the original object. Some objects are easier to describe by their horizontal cross sections, while others are easier to define by their vertical cross sections. The number of sections that are necessary to describe an object accurately can range from half a dozen to more than a hundred. A loss of data is apparent

when the skinned objects are sampled at large intervals and the cross sections are too far apart from each other. Once the cross sections have been digitized, it is sometimes necessary to edit the database to adjust individual sections within the three-dimensional object. Geometric transformations can be performed on the contours to change their relative spatial position.

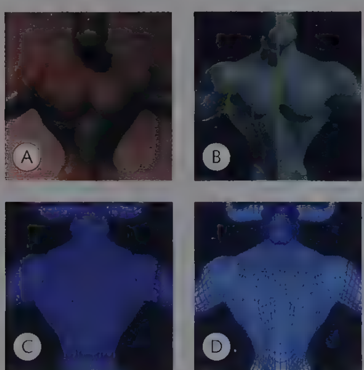
The contours used for skinning the outside and inside shapes of human bodies can also be gathered automatically and precisely with a variety of technologies, such as three-dimensional laser scanners and magnetic resonance (MR) scanners (Fig. 5.1.5). The three-dimensional data used to create the stereo pair of a human thorax illustrated in Figure 5.1.6, for example, comes from a person who did 45-

5.1.7 The characters in the *Four Hoursemen* were each modeled separately then rendered in multiple passes (color, ambient occlusion, and specularity) within the same ZBrush scene. The tentacles were created with the ZSphere tools, then converted to polygons to sculpt detail. The flames are a fluid dynamics simulation in Maya, tweaked and retouched in Photoshop. The painted background was composited with a Z-Depth pass. (Image courtesy of Meats Meier.)



5.1.8 Detail of a rendering of characters created with blobby surfaces, and a wireframe geometry detail of the main character. (Image courtesy of Tim Cheung.)

second breath holds in a magnetic resonance (MR) scanner. At each breath hold, a series of scans of a single coronal slice were taken at ten different phases of the heart cycle. This was achieved by coordinating the scanning process—or data acquisition—with the electrocardiogram signal of the subject. This process was repeated about 50 times at different cross sections of the heart. The complete acquisition took about an hour, and it yielded 490 slices, comprising ten groups of 49 slice-volumes at ten different phases of the heart cycle. The flowing blood appears as the brightest object in the images.



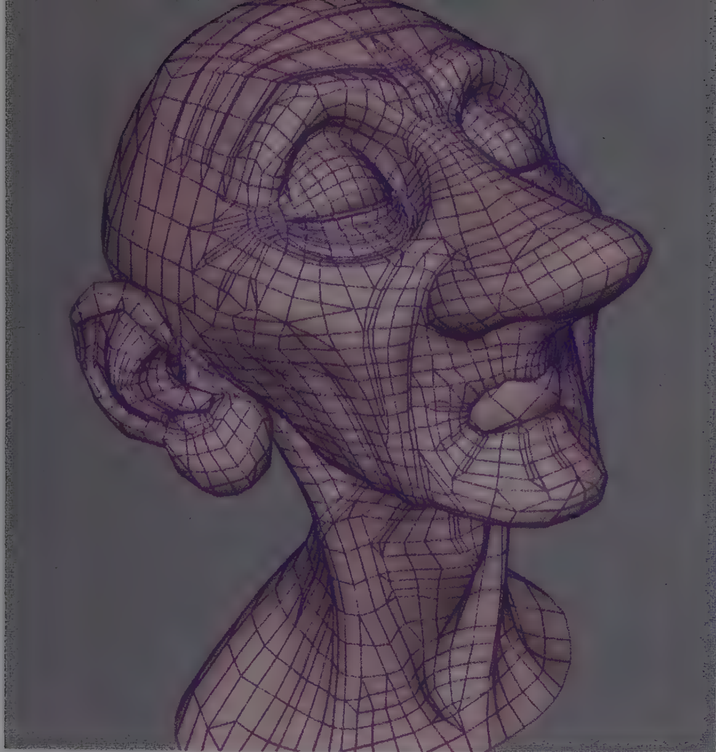
5.1.9 The modeling process of fitting curved surfaces to a polygonal mesh starts here with (A) a scaled sculpture that was digitized with a laser scanner. After the raw scans are integrated control curves are created (B) to generate the fitted NURBS curved surfaces (C-D). (Images courtesy of Domi Piturro.)

Blobby Surfaces

Blobby surfaces are usually defined as spherical objects that change shape depending on how close they are to other blobby elements. The magnitude of the **attraction force** of blobby elements is usually defined by their volume, but their **area of influence** can also be set independently from their size. In an animation, **blobby surfaces** are dynamic and constantly regenerate as they move in and out of the areas of influence of other blobby elements (Figs. 5.1.7 and 5.1.8). When two or more of these touch one another they can fuse into a single object, just as two drops of oil or mercury merge into one another when they meet on a surface. The way in which blobby objects, also called **implicit surfaces**, fuse into each other is determined by the explicit or random links that are established between each of the objects in a blobby system.

Fitting Curved Surfaces to Polygons

The techniques of fitting curved surfaces to polygonal meshes have gained popularity due to the increased use of three-dimensional scanners to capture physical models and maquettes. Scanners usually



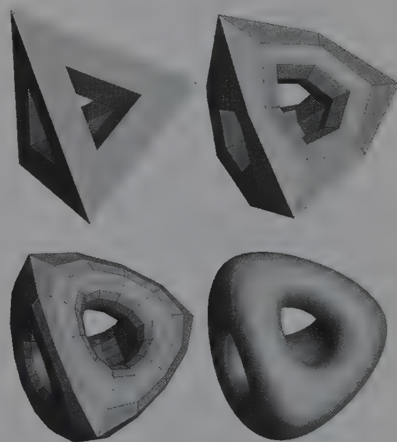
5.2.1 The short film *Geri's Game* was a technology test bed for subdivision surfaces and clothing dynamics. The model of Geri's head shows the adaptive point density typical of subdivision surfaces. The vertices where more local detail was desired were selected, and then the subdivision procedure was applied to them (see rendered version in Figure 1.2.9). Each hand (next page) is a single seamless subdivision surface with a few sharp edges, which are obtained by subdividing just along the edge but not across. Notice the sharp crease along the thumbnail, and several softer creases with variable sharpness as it moves around the area. (© Pixar Animation Studios.)

capture three-dimensional geometry in the form of polygonal meshes, and specialized software has been developed to help convert the polygonal information into curved surfaces. The purpose of this type of software is to convert the data in an efficient way but also to allow a human modeler to specify how the patches should be positioned and oriented on the polygonal mesh (Fig. 5.1.9).

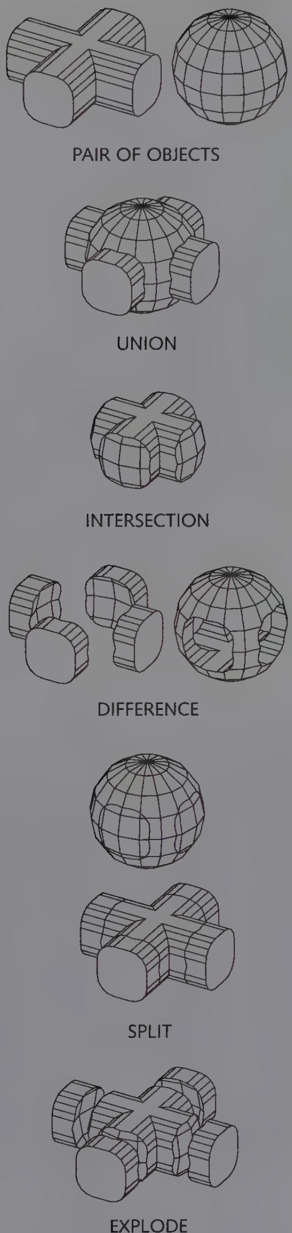
5.2 Subdivision Surfaces

Subdivision surfaces are popular as a flexible solution for modeling surfaces. These surfaces are something of a cross between polygonal meshes and patch surfaces, and they offer some of the best attributes of each of these traditional modeling techniques. Subdivision surfaces have the flexibility of polygonal meshes but without the faceted look typical of low-resolution polygonal geometry (Fig. 3.6.5). Subdivision surfaces can also yield smooth curved surfaces without the topological restriction of patches where the number of columns and rows has to be identical before two patches can be merged with one another.

Instead of having a constant density of points throughout the model, subdivision surfaces allow for different resolutions on arbitrary sections of the surface. Subdivision surfaces inherited their property of subdivision from curved patches, but they excel at modeling creases and cracks that are difficult to manage with b-spline curved surfaces, for example. But subdivision surfaces are not parametric (as curved surfaces are) because their topology is irregular



5.2.2 The subdivision process splits each surface into four facets by adding edge midpoints and face centroids. (© Pixar Animation Studios.)



5.3.1 Logical operators applied to a pair of objects result in a basic union, an intersection, and two views of the difference. The difference operator is often used to split objects, and to explode the new objects into parts.

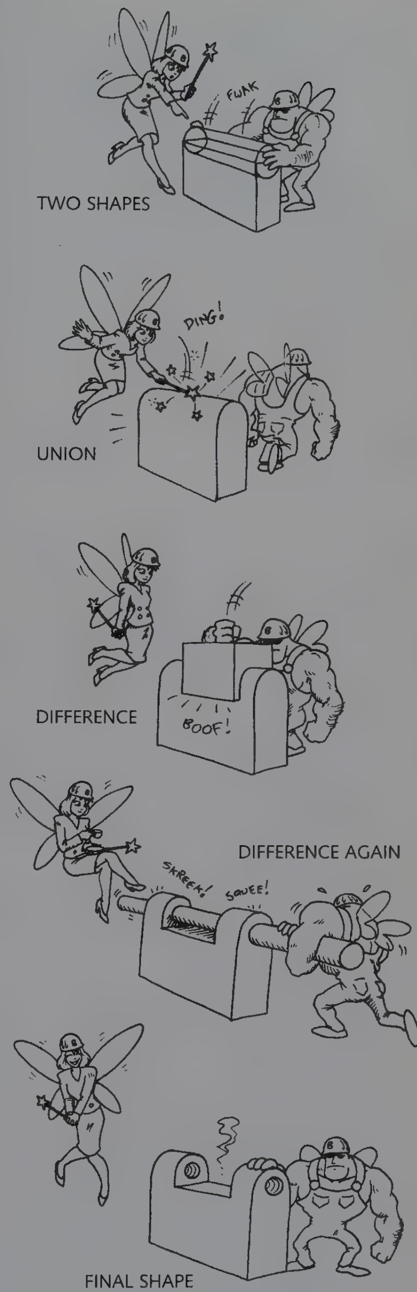
and is not defined with a formula in an explicit way. Subdivision surfaces are defined algorithmically, and many of the algorithms that produce more polygons do so in two steps: first, split each surface into four facets and then reposition the vertices by doing **local weight point averaging**. As shown in Figure 5.2.2 this procedure can be applied multiple times to create more detail. There are many ways to go about subdividing a surface including interpolation, averaging, approximation, and insertion of new points. But for these approaches to be efficient, they are usually based on **adaptive approximation**, which means that the surface will subdivide only where the topology of the surface requires additional detail. Subdivision surface techniques were first used in production in *Geri's Game* and *A Bug's Life*. Each of Geri's hands, the head, the jacket, and the pants was a single subdivision surface (Figs. 5.2.1 and 1.2.9). The faces and the hands of Flick and Hopper were also modeled with subdivision surfaces, while the sets were done with curved surfaces. Image maps are best placed onto the subdivided surfaces when the texture coordinates are subdivided in texture space exactly like the models were subdivided. The traditional UV techniques for mapping image maps onto subdivision surfaces do not always yield accurate results.

5.3 Logical Operators and Trimmed Surfaces

Logical operators are used to create models by adding and subtracting shapes in a variety of ways. The most common logical oper-

ators include **union, intersection,** and **difference.** The union operator is also known as **and,** intersection is called **or,** and difference is called **not.** The union and the difference of a sphere and a cross are illustrated in Figure 5.3.1.

The difference logical operator is usually referred to as **trimming**, and the surfaces created with it are called **trimmed surfaces**. The modeling technique of trimming is especially useful for creating three-dimensional objects or **surfaces with holes**. The sequence illustrated in Figure 5.3.2 shows this operator in action along with the union operator. The union of a cylinder and a box results in a box with a rounded top. The difference between the rounded box and a cube yields a rounded base with a rectangular trim. Finally, the difference between the base and a cylinder results in a base with two round holes. Trimming requires at least one pair of intersecting objects as illustrated in Figure 5.3.1. But others can trim surfaces or objects with just a line or a two-dimensional shape—the hole—that intersects the shape or that is projected through the object.



5.3.2 This sequence illustrates the logical operators of union and difference being used to model a complex three-dimensional object from four components.

5.4 Advanced Modeling Utilities

Many modeling utilities like beveling, fitting, blending, and spline deformation are used to further manipulate surfaces, and to complement the polygonal, curved, and subdivision surfaces techniques.

Beveling, Rounding, and Fillets

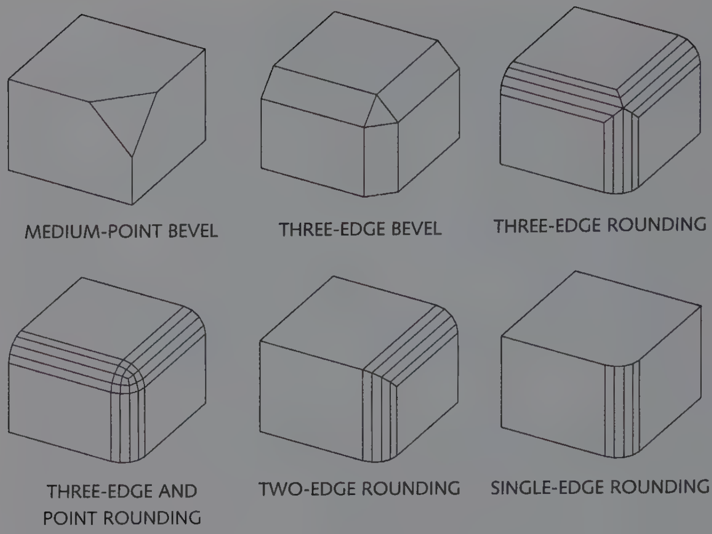
The edges between adjacent surfaces can be customized with great detail with a variety of beveling techniques. Simple **beveling** usually works by truncating the hard edge between adjacent surfaces—usually two or three—and replacing it with a slanted plane. The amount of beveling can be controlled by a distance, radius, or angle value between the edge being beveled to the place where the bevel is to be placed. **Rounding** is a delicate form of beveling that literally rounds the straight edges or points of an object. The degree of rounding is controlled by the number of segments or facets that are used to define the smooth transition between adjacent surfaces (Fig. 5.4.1).

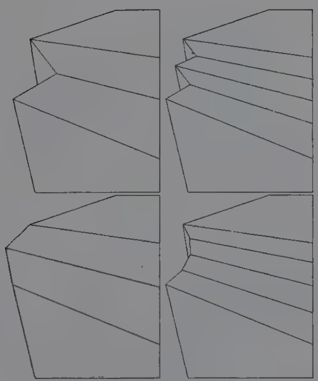
Utilities for creating fillets also modify the junction of surfaces. **Fillets** create a custom trim that extends along the edge. This trim is more ornate than plain beveling or rounding. Fillets are like the decorative strips of molding that are often placed at the corners or edges of furniture, or like the functional molding that is used to protect the edge where walls meet the floor. Some software programs create fillets by trimming the surface, and others by sweeping a custom two-dimensional outline along the edge that is being modified (Fig. 5.4.2).

Aligning, Fitting, and Blending

Utilities like aligning, fitting, and blending are especially useful for fine-tuning the edges of curved patches—and can also be used on

5.4.1 The first shape shows a simple point bevel. The next three shapes display a three-edge bevel, a three-edge rounding, and a three-edge and point rounding. The last two shapes are examples of a two-edge rounding and a single-edge rounding.



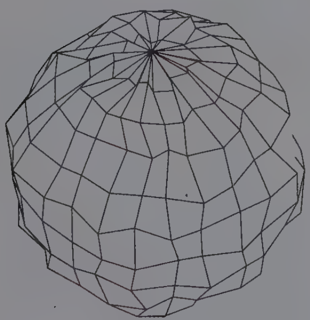


5.4.2 This figure shows the results of four different fillet styles applied to the corner and edge of a cube.

polygonal surfaces. **Aligning** two patches usually works by selecting the two patches to be aligned and then moving and rotating them until they are aligned a certain way. **Fitting** utilities get rid of small gaps between surfaces by dragging the two surfaces that are not quite touching each other until their edges match each other perfectly. Fitting is almost always used prior to merging two patches. **Blending** is a special way of merging two surfaces. Instead of merging two surfaces by first making them touch each other and then merging them, blending creates a new surface that extends from each of the two surfaces being blended. The new surface created by blending connects the two surfaces, and the smoothness of the blending is controlled with a function curve or by manipulating the control points of the blended surface (Fig. 5.4.3).

Purging Points

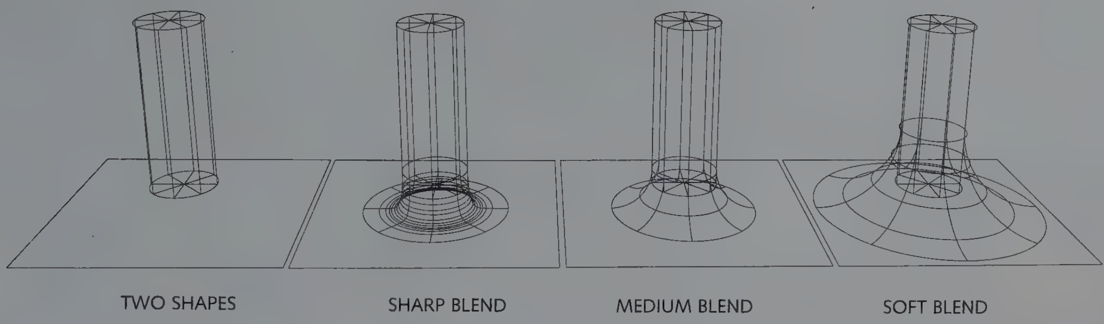
Three-dimensional models created with curved patches often have a large number of points or vertices. This results in models that may be too complex for the requirements of the project and also increases rendering time and file size. **Purging** utilities are useful for automatically eliminating excessive vertices in complex three-dimensional models. This is usually done by identifying pairs of points that are too close to each other—based on a minimum distance—and deleting one of them. Manual point-editing is often used in conjunction with purging utilities to fine-tune and adjust the distribution of points in the model.



5.4.4 Random distortion of a sphere.

Deformed and Randomized Surfaces

In addition to using lattices for deforming the shape of three-dimensional objects, it is also possible to deform them with splines, patches,



5.4.3 Blending a cylinder and a plane with different degrees of transition between the two surfaces.

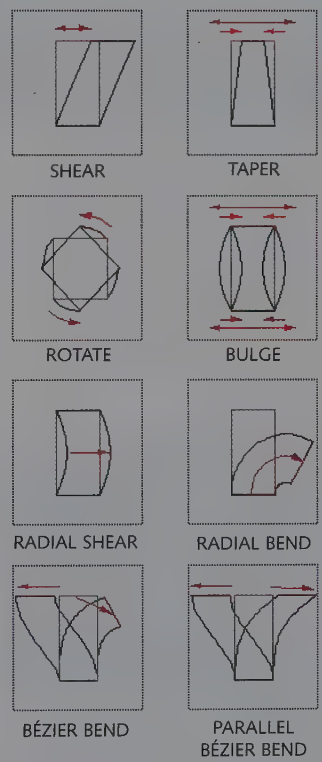
functions, or random numbers. (Deforming objects with lattices and functions is covered in Chapter 4.)

The technique of **deformation with splines and patches** consists of using a spline or a patch as the agent that deforms the object that is associated with them (Fig. 5.4.5). Deforming an object with a spline offsets the points in the object according to the shape of the spline. Deforming an object with a patch pulls the object to the shape of the patch in those areas where the object overlaps the patch.

Interesting deformations can be achieved by offsetting the vertices of three-dimensional objects with functions or random values (Figs. 4.5.4 and 5.4.4). The technique of **random distortion** is especially useful for creating models of terrains that have so many irregularities that it would be difficult to model them with other techniques. Random distortion can also be used to animate the effect of shaking by animating the displacement of points back and forth through time.

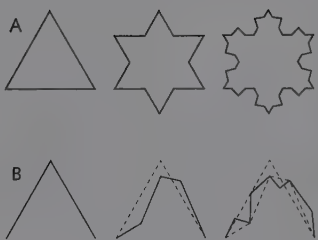
5.5 Procedural Descriptions and Physical Simulations

Procedural descriptions of three-dimensional models—especially those found in nature—are effective alternatives to the regular and sometimes rigid shapes obtained with geometric modeling systems. With procedural description objects are not modeled by sculpting their exterior shell. The modeling techniques of **procedural description** get their name from the fact that, with them, objects are modeled by simulating, for example, their natural growth process that is described in the form of procedures (Figs. 5.5.6–5.5.9). Fractal geometry and particle systems, for example, are two procedural description methods that create a modeling complexity that is difficult to obtain with geometric modeling. Both of these methods are well suited for the generation of natural-looking forms because they allow for randomness, recursion, and accidents of shape like those typical of natural shapes. A wide variety of plants can also be described with a combination of procedural description techniques. Finally, the shape of many natural phenomena that we do not think of as having a three-dimensional shape—waves, clouds and smoke, for example—can be built by simulating their physical behavior.



5.4.5 Sequence of icons illustrating different techniques for controlling patch deformation. (form•Z icons, © 1991–1995 auto•des•sys, Inc.)

5.5.1 Detail from *Seasons of Life* illustrates how fractal procedures, used to model the trees' branches and leaves, can express the roughness and fragmentation of natural phenomena. (Courtesy of Midori Kitagawa.)



5.5.2 A fractal recursive subdivision is applied to a regular polygon (A), and to the detail of a three-dimensional mesh (B).

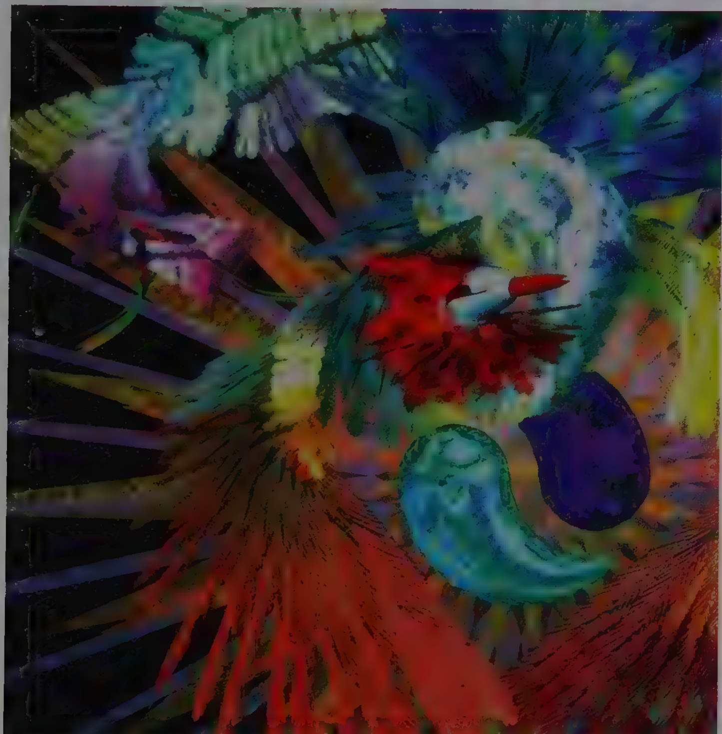
5.5.3 (Opposite page, top) A fractal pattern reminiscent of a stained glass was formed from the reflections created by four intersecting spheres inside a tetrahedron. Ray-traced with six levels of reflection. (© 2008 Duncan Brinsmead.)

Fractal Geometry

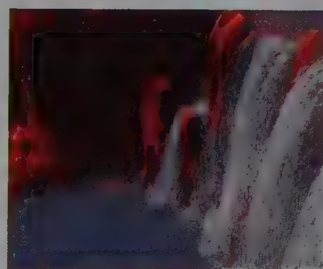
Fractal geometry is especially effective for creating random and irregular shapes that resemble shapes found in nature (Fig. 5.5.1). This modeling technique was developed by Benoit Mandelbrot in the 1970s. It can be applied to existing three-dimensional meshes, or it can be used to generate entirely new models or parts of models (Fig. 5.5.3). When applied to an existing model, **fractal procedures** work by dividing the polygons in the object recursively and randomly into many irregular shapes that resemble those found in nature. The amount of subdivision is usually expressed in the form of a factor or **level of recursion** (Fig. 5.5.2).

Particle Systems

Modeling with **particle systems** is based on employing simple shapes, usually small spheres or points in three-dimensional space. These shapes, or particles, have **growth attributes**. When these attributes are simulated, the particles have specific behaviors that result in specific particle **trajectories**. As the particles grow over time their trajectory defines a certain shape that results in a three-dimensional model. The growth process of many of the attributes of the particles, including their height, width, branching angle, bending factor, number of branches, and color, can be controlled or randomized. Figure 5.5.4 shows a scene that was modeled by simulating

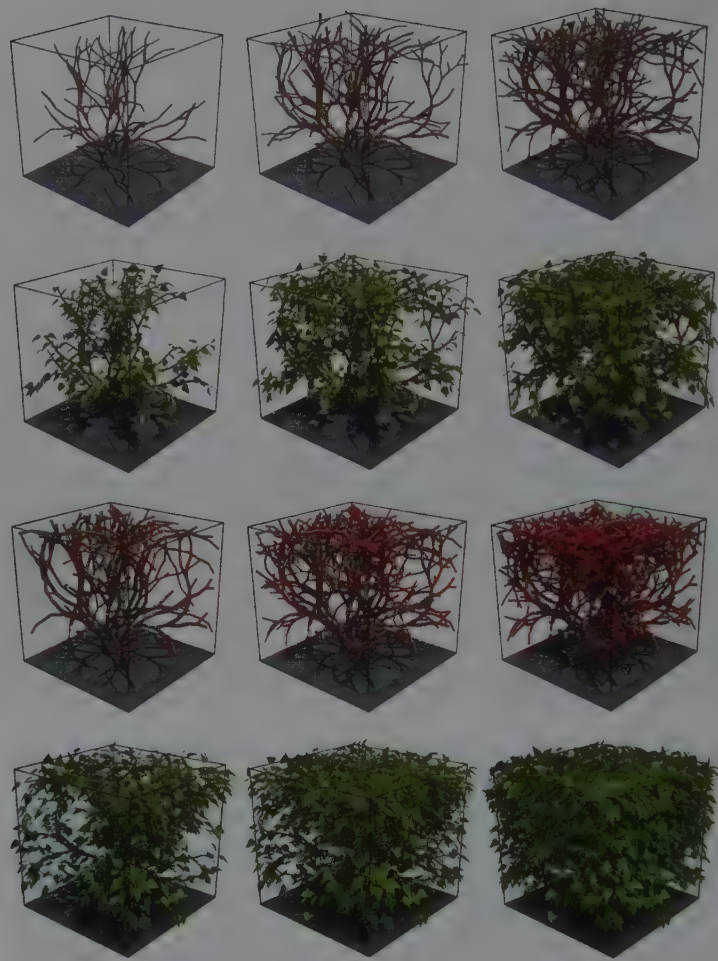


5.5.4 (Left) The feathers in *The Holy Bird* are created with particles. The animation is based on the traditional dances of Okinawa. (Production: Digital Studio, Inc. Assistant designer: Atsuko Katakura. Courtesy of Ryoichiro Debuchi.)



5.5.5 This still from the animation *Flow* shows a particle system combined with a water mesh simulation system. (Rendering and animation by Gavin Miller, modeling by Ned Greene. © Apple Computer, Inc.)

5.5.6 These plants clearly show the transfer of attributes, called lineage, from one level of the plant to another at the time of branching. (From "The Algorithmic Beauty of Plants," by P. Prusinkiewicz and A. Lindenmayer, Springer Verlag, New York, 1990. © 1994 Przemyslaw Prusinkiewicz.)



5.5.7 (Top right) Simulated response of a tree to progressive pruning. The branches grow more densely near the edges of the cube used to clip the model. The leaves are defined as Bézier surfaces. Shown in red is the vigor of the reiterated branches on the appearance of the pruned tree. (© 1994 P. Prusinkiewicz, M. James, and R. Mech.)

feathers of birds or petals of flowers by creating particles that grow densely on the surfaces of three-dimensional objects. In this project, colored maps were applied to the feathers, the bird is animated with a skeleton deformation technique, and all the feathers grow at their positions as the body moves. (Read Chapter 12 for more examples and information about animating with particle systems.)

Procedural description, and particle systems in particular, are used to create a variety of natural materials and phenomena whose shape constantly changes throughout time. This includes, for example, snowstorms, clouds, flowing water, moving soil, and fire. When particles are used to recreate the motion of water, for example, they each represent a drop of water with attributes like density, cohesion, transparency, and refraction. Particles have a life span during which they behave a certain way, and then fade away or merge with other particles (Fig. 5.5.5). Of all the procedural modeling techniques, particle systems is the one that best recreates the dynamic shapes of natural

[illegible]

Plant	Objects	Instances
Apple trees	1	4
Reeds	140	140
Dandelions	10	55
Grass tufts	15	2577
Stinging nettles	10	430
Yellow flowers	10	2751

5.5.8 The images representing each species of plant in the top image were rendered separately and later composited. The table lists the number of prototype objects and their instances for each species. The compute time on a 195 MHz R-10,000 8-processor SGI Onyx with 768 MB of RAM was 75 minutes. The poplar tree in the top image is 16KB when saved as a procedural model but 6.7 MB in a standard text geometry file format. (Copyright Bernd Lintermann. Software by Oliver Deussen and Bernd Lintermann.)

elements. This technique produces large numbers of particles that do not have a specific shape and are usually spheres or dots. But the particles are grouped in shapes that change through time according to rules that define the behavior of living organisms and elements such as water and fire (Figs. 5.5.9). Particle systems are a popular and powerful technique for the animation of effects both in animated and live action movies (Figs. 12.3.4, 12.4.1, and 13.7.1).

Modeling Plants

Three-dimensional models of plants and trees created with procedural techniques offer increased modeling control and more efficient modeling than most other techniques. This is due to the large number of elements and surface detail contained in a plant, as well as the complexity of shapes. Modeling plants directly with polygonal and curved surfaces modeling techniques is possible but usually time consuming. Some software programs, Maya's Paint Effects for example, offer solutions that combine interactive painting with particles or with three-dimensional geometry. This method allows you to

5.5.9 (Next page) This organic form was "sculpted" by growing a mass of particles over time. Simulation software traces paths for millions of particles within a force field, creating forms as the particles flow and settle. The image was rendered as implicit surfaces using proprietary ray tracing software. (Copyright Andy Lomas.)

paint textures in three-dimensional space and to employ leaves, flowers, and grass as brushes that create three-dimensional strokes along a NURBS curve. This approach allows the creation of quite complex environments with geometry that appears to grow in a natural way (Fig. 5.5.10).

Models of plants can be built by encoding their characteristics in a series of rules or procedures that are used as the basis for a growth simulation. This method can also be used to animate the growth process of plant models in a scene. There are several ways to generate models of plants with procedural techniques. These general techniques can be classified as either space-oriented or structure-oriented.

Space-oriented procedural techniques for modeling plants and animating their growth are based on the effect of the environment on them. **Structure-oriented procedural techniques**, on the other hand, are based on the conditions that are internal to the plant, more specifically the growing process of the plant and the resulting structure that is characteristic of a plant species. Many of the procedural modeling techniques centered on the growth process of plants are often expressed in terms of the mathematical models formalized by biologist Aristide Lindenmayer during the late 1960s. These structure-oriented models describe the growth process of plants at the level of cellular interaction and are known as **L-systems**, short for Lindenmayer systems. L-systems are especially suited to represent structures that **branch in parallel**, just like a tree trunk splits into several branches at the same time (Fig. 5.5.6). Another characteristic of L-systems is that at each branching cycle, the branching procedures generate successor modules that replace the predecessor modules.

There are many variations of L-systems. Some are defined by parameters that represent exclusive conditions under which the L-system can bloom. Others are based on **stochastic** (or somewhat random) **values**. **Context-sensitive L-systems** are those whose performance is defined by the characteristic of the preceding module. The technique of **graftals** is an example of context-sensitive L-systems that allows the creation of complex images from small databases, a technique known as **database amplification**. Graftals are based on production rules and generation factors; they avoid random number generators, and can employ geometric or nongeometric objects—spheres, cylinders, or fuzzy objects with smooth edges.

Environmentally sensitive L-systems are defined by environmental characteristics, such as exposure to light and collision with objects. Pruning is an example of three-dimensional models based on a simulation of a growing plant whose shape is determined by an environmental variable (Fig. 5.5.7).

Some of the main techniques for controlling the simulation of a structure-oriented plant system include lineage and the interaction between the components of the growth process. **Lineage** is the transfer of attributes from one level of the plant to another at the time of branching. The interaction of the components of the growth

5.5.10 The plants in *Mother Nature Summer* were created with Maya's Paint Effects, and rendered in layers. See Figure 4.5.7 for another image from the same series. (Image courtesy of Meats Meier.)

process includes, for example, taking into account the watering conditions or the availability of nutrients that define the way in which plants grow. The plants modeled with procedural description are more faithful to the real plant based on the number of components that are considered in the derivation of the plant shapes.

Techniques that seek to find efficient ways to process the large amount of data necessary to model and render realistic plants are actively explored. The obvious challenges include dealing with billions of geometric primitives, distributing them throughout the terrain

5.5.11 Between 30,000 and 60,000 strands were used to create Aki's hair, which accounted for 25% of the total rendering in *Final Fantasy: The Spirits Within*. The hair appears in 611 shots, almost half of the total. Aki's body and head model has up to 100,000 subdivided polygons. (© 2001 FFFP.)

5.5.12 Fabrix simulation software was used to shape the purple fabric and the gold backdrop. Clusters were positioned and scaled to gather a 20 × 40 NURBS mesh into the hands of the high-resolution human modeled by Zygote. The simulation also tested for self-collisions. (*Fab-girl* by Shawna Olwen. © 1999 Reflection Fabrix Inc.)

in a realistic way, and rendering all the subtle light and shadows effects that one observes in natural landscapes. Figure 5.5.8 illustrates the result of a technique that can populate the environment with a combination of ecosystem simulation and gardening techniques where the location of plants is done by hand. The figure also contains the statistics of instancing and geometric compression for the scene. This technique also makes extensive use of **model approximate instancing**, where plants or groups of plants are replaced by instances of representative objects before the scene is rendered. In exploring complexity, the researchers behind this technique found out that when one of the scenes approaches 100,000 plants each plant is visible on average only in ten pixels of a 1000 × 1000 pixel image.

Modeling with Physical Simulations

The easiest way to model natural materials whose shape is in constant change is not to sculpt them but to simulate them. Physical simulations are used extensively to model natural phenomena like rain, fire, smoke, and even wind; some of those techniques are covered in Chapter 12. A grid of waves may be animated with the basic variables of average wave height and slope, direction, frequency, and speed. Other global settings determine the complexity of the waves, the wave patterns, and underwater shots. Secondary motion in the simulation can be controlled at different levels of detail in the form of ripples, swells, and boat wakes (Fig. 11.2.8). The shading parameters in a physical simulation of water may include variables such as glitter, mapping of two-dimensional image and bump maps on the surface, reflection blurring, and index of refraction. Depth and visibility fading is often used for underwater scenes. Figure 5.5.14 shows the result of a daytime ocean simulation. Some of the variables typi-

cally employed in the simulation of water surfaces include: wave height., wave angle direction, speed, level of wave complexity, wave slope, and geometric density of mesh. Figure 12.3.6 shows a simulation of pouring water on a car, and Figure 12.3.4 shows clouds.

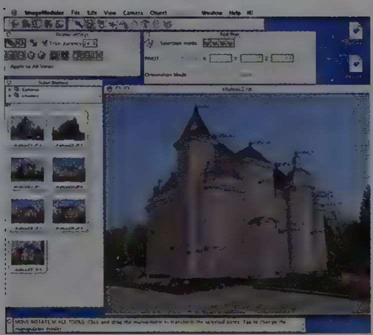
The shape of solid but flexible objects can also be controlled with physical simulation software tools. Figure 5.5.12 shows fabric whose shape has been defined by a simulation of gravity and other forces that affect them. Natural-looking hair is usually simulated with multiple strands of particles that are guided by a few control hairs (Figs. 5.5.11 and 5.5.13).

5.5.13 Sully from *Monsters, Inc.* used 10,000 control hairs to define the placement, orientation, and animation of between 2.8 and 3.2 million blue hair strands. (© Disney Enterprises, Inc./Pixar Animation Studios.)

5.6 Photogrammetry and Image-Based Modeling

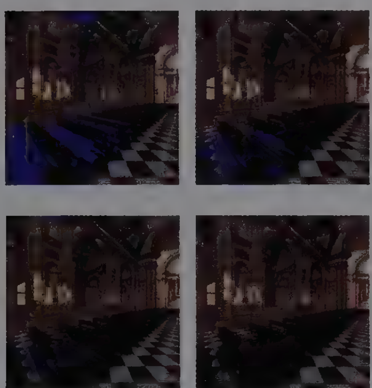
Photogrammetry techniques allow us to reconstruct three-dimensional environments by extracting spatial information from a series of photographs of an environment from known locations (Fig. 5.6.1). The reconstruction process in **photogrammetry** programs starts by indicating in the scanned photographs the corners or main edges of the major shapes (Fig. 5.6.3). Photogrammetry software reconstructs a simple version of the geometry by analyzing and comparing perspective lines and shading information between related photographs of the same environment. Figures 6.8.1 and 6.8.2 provide a stunning example of image-based modeling, and Figure 6.12.3 show a reverse process used to locate photographic views in three-dimesional environments. **Laser scanning** is another technique that can be effective in modeling large-scale three-dimensional environments. This technique often uses a linear laser scanner to collect three-dimensional point data which is then converted into a polygonal mesh. An interesting use of laser scanning is coupled with traditional film or video cameras to capture high-resolution image maps that can be applied to the geometry (Fig. 5.6.2).

5.5.14 Surface of the sea modeled by simulating the motion of the water mass and water surfaces. (Courtesy of Areté Entertainment, Inc.)



5.6.1 (Above) Dialog box from Image Modeler, a software that allows three-dimensional reconstructions from photographs. (© Realviz®.)

5.6.2 (Top right) The *Panascan*™ laser scanning process was used to capture the geometry of a helicopter and the environment surrounding it. (Courtesy of Panavision.)



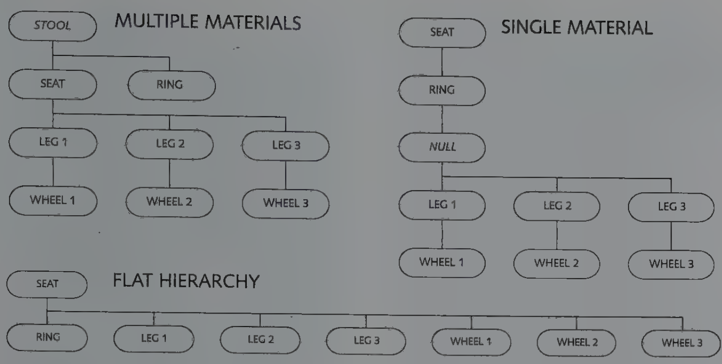
5.6.3 The top left view shows some of the geometric detail (shown in blue) that was not captured in the original photographs. The rest of the views show steps in the process of approximating the data and filling it in with the reconstruction software. (Courtesy of Chun-Fa Chang, G. Bishop, A. Lastra, N. O'Brien, and ACM.)

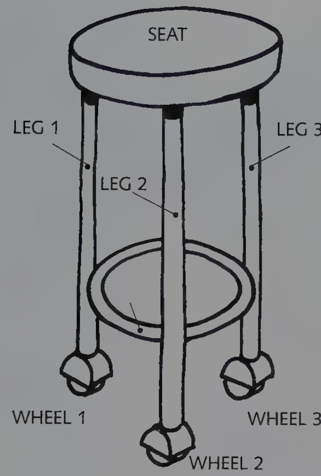
5.7 Animation Rigging and Hierarchical Structures

Three-dimensional objects can be grouped together in a limitless number of ways in order to create structures that define the ways in which these models behave when animated, how they relate to each another, and how they are rendered. Groups of three-dimensional objects are usually called **hierarchical structures**, because within these structured groups some objects are always more dominant than others. (See Chapter 11 for additional information on levels of precedence, joints, and degrees of freedom.) Hierarchical structures are the most basic layer of an animation rig.

Objects within hierarchical structures have defined levels of importance and the objects within this hierarchy inherit attributes from the dominant objects. The object or objects at the top of the hierarchy are called **parents**, and the objects below are called **children** and grandchildren: Children inherit their parents' attributes. Hierarchical structures can also be visualized as an inverted tree structure where the highest level of importance in the structure corresponds to the trunk of the tree. The branches that come out of the trunk are the next level of hierarchy, branches that come out of the main branches are the next level, and so on until we get to the leaves, which are in the last level of the hierarchical structure.

In most cases, there is just one set of hierarchy diagrams per scene, and these diagrams control all the modeling, rendering, and animation information about the objects. Some programs, however, provide several sets of hierarchy diagrams, and therefore hierarchy configurations, so that **modeling links** can be independent and different from the **rendering links** and the **animation links**. Modeling links, for example, make sure that the shape of all the children throughout the hierarchy changes when the shape of the child's par-





5.7.1 The diagrams show three possible variations of the hierarchical structure for all of the stool's components. The first arrangement (top left) would be the most practical one if the stool were to roll on its round wheels, and if the seat, legs, and wheels were each to be rendered with a different material. The second option (top right) would be best if the entire stool was made out of just one material. The third option (bottom) has the seat at the top of the hierarchy and everything else below the seat at the same level.

ent is modified. Rendering links transmit the rendering attributes of the parent down the structure. Animation links automatically update the spatial position of the children when the parent is transformed.

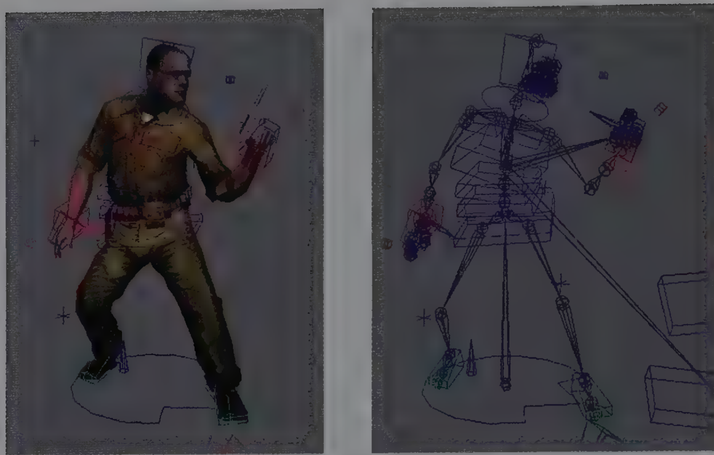
The relations between objects in hierarchical structure are often complex and can be better visualized by a schematic representation in the form of a line diagram. These diagrams are often built with boxes representing items in the structure and lines representing their place in their hierarchy and their relations with other items.

As shown in Figure 5.7.1, there are many ways to group objects or nodes in a hierarchical structure. The best choice of hierarchy is usually one that takes into account the rendering and animation requirements of the scene. (See Chapters 11 and 12 for more details on how hierarchical structures affect the object's rendering and transformation attributes.) A node in the hierarchy that does not contain an object but holds several children together is called a **null parent**, or null for short. A null parent is used, for example, when two or more objects are grouped at the same level in the hierarchy. Nulls are often represented as empty boxes in the structural diagrams. Figures 5.7.1 and 5.7.2 illustrate the use of a null cell that can be used to keep the three legs and the wheels in one group so that they can be manipulated independently of the ring.

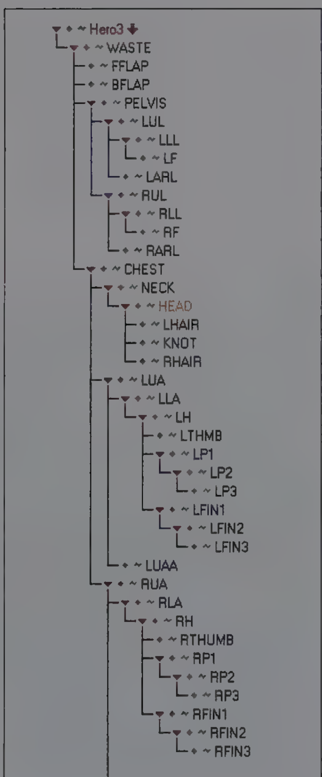
The specific steps required to establish hierarchical relationships between objects vary from software to software. But the basic concept has two variations; clicking on the three-dimensional objects themselves in any one of the camera views, or clicking on the boxes representing the objects in the diagram that shows the links between them. Some programs require users first to click on the object that is to be the parent and then on the children. With other programs the object that is to represent the children must be clicked before the parent. Hierarchical structures are more significant to the stages of rendering and animation, they are usually created during the modeling process in the form of animation rigs.

An animation **rig** is a hierarchical control structure that is custom-designed to animate characters. The animation rig is the motion engine of the character; all the joints and all the motion logic of the character are contained in the rig. Very complex rigs require signifi-

5.7.2a A portion of the internal hierarchy of this real-time animated character grasping an object is represented by the diagram on the next page. (© Motion Factory.)



5.7.3 Animation rig of the gunnery sergeant in *Medal of Honor*. Notice the animation control handles used by animators to pose the character. Two of his core poses are shown in Figure 10.4.16. (© 1999 Electronic Arts Inc. All rights reserved.)



5.7.2b Notice how the root of the diagram controls the pelvis and the chest. The pelvis drives the upper and lower legs, and the feet. The chest drives the neck, the upper and lower arms, and the hands. (© Motion Factory.)

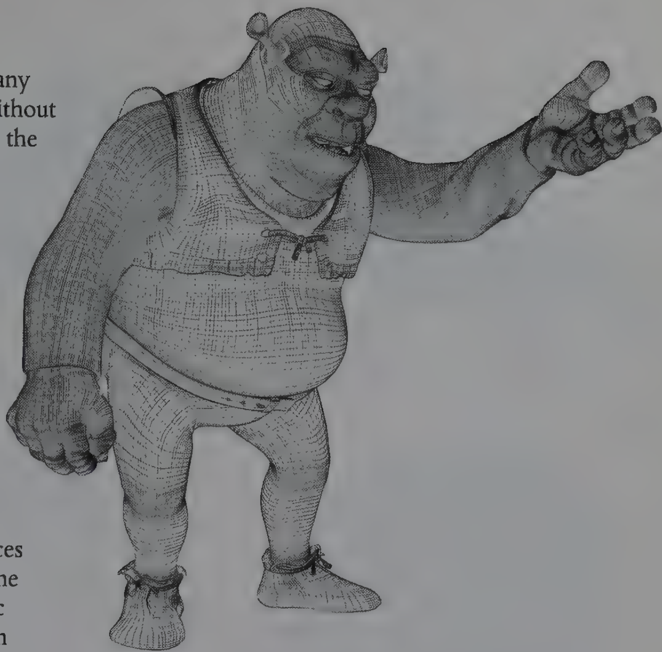
cant computation and when manipulated by animators they take longer to update, so it is best to include in only the essential features. In most large-scale projects control rigs are usually designed and assembled by specialized technical directors called riggers. For a small production with limited resources a simple rigging and skin binding structure can be more efficient and minimize rendering times.

Animation rigs can be visualized in different ways: as a hypergraph or hierarchy chart (Fig. 5.7.2b), as a rig skeleton showing bones and joints (Fig. 5.7.3), or as the set of controls that animators use to manipulate the rig system. These animation controls are represented with icons and color-coded (Fig. 12.5.6) so that animators recognize faster what controls do or what parts they control. During the animation process the rig skeleton is usually turned off as animators pose the character with a proxy, low-resolution version of the skin and with the animation handles (Figs. 5.7.3 and 5.7.9). Sophisticated rigs are capable of notifying the animator through visual cues when a limit is reached or when the animator is trying to do something that the character is not supposed to do, like trying to bend a knee backwards or pass a foot through the ground.

Animation controls in a rig can consist of a point, a curve, or a complex surface. For ease of use animation controls often include multiple controls—a foot control, for example, might include foot pivoting, toe lifting, wiggling, and twisting. The legs and feet are crucial in animating a character since they are so important in walking. Most leg-feet rigs include joints for hip, knee, foot, ball of the foot, and toe end (Fig. 12.1.9). Facial animation rigs also include multiple bones as shown in Figure 12.5.5.

One of the basic issues in setting up a rig is defining the order in which rotations take place. The **order of rotations**, as explained in Chapter 3, is important because the sequencing of axis rotations creates different results (Figs. 3.4.1 and 3.4.2). Software programs usually require an order of rotations to be specified in a rig, with the Y axis commonly being first, followed by X or Z in worlds where Y defines

the vertical axis. This way characters can turn in any direction around Y and move sideways or lean without reaching **gimbal lock**, which is a point at which the model runs out of rotations.



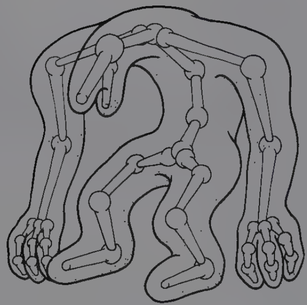
5.7.4 The skin model of *Shrek*, including clothing. Figure 5.7.5 shows some of the muscle and bone rigging system used to animate the skin surface of the character. (*Shrek*™ and © 2001 DreamWorks L.L.C.)

Binding the Skin

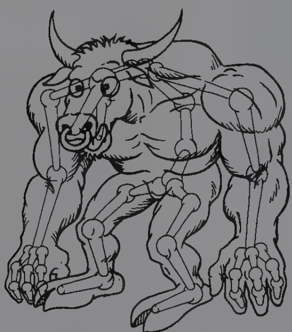
The surfaces covering the skeleton are called **skin surfaces** because they are continuous and flexible like skin—they deform in response to the movement of the skeleton that they cover. The process of attaching the surfaces of a model composed of polygonal meshes to an **articulated skeleton** is known as **skin binding**. Skin surfaces may be attached to an animation rig by positioning the surface over the skeleton and manually attaching groups of vertices to skeleton nodes (Fig. 5.7.6). This is usually done by linking specific areas on the surface to specific chain segments. Once the skin surfaces have been assigned to different segments of an articulated chain or skeleton they can be further refined and controlled by specifying their **deformation parameters**. These parameters allow animators to control the various properties of bending skin and to manipulate single vertices or groups of vertices on the skin's surface. The deformation parameters include the amount of bulging or rounding of a portion of the skin surface as a result of a joint bending. These parameters also control the weight that defines how the vertices will respond to deformation, the way adjacent regions blend with one another, and the assignment of vertices on the skin surface to a specific segment in the articulated chain (Fig. 5.7.8). Optimizing the way in which the surfaces deform may be a laborious process of trial and error. It is early during this process when areas on the surface lacking in detail are identified so that more polygons can be added in places—like the joints—that will bend under skeleton control. Having the right amount of geometry minimizes unnatural stretching effects or extreme distortion of the skin (Fig. 5.7.4).

Ideally the geometry to be attached to an articulated skeleton should be built with a topology (shape and structure) that does not change radically when scaling or rotations are applied. Good topology will naturally follow the deformations indicated by the motion of the **articulated chain** segments. When bound properly to the animation rig, skin surfaces may automatically adjust to modifications in the translation, rotation, and scaling of the skeleton. The realistic effect of skin that stretches with motion—especially at the joints—is often achieved only with a combination of animation techniques that may include motion dynamics of flexible bodies, attaching the skin to the bone and muscle systems using spring-like simulations (Fig. 12.1.7), and free-form shape animation in the form of flexible lat-

5.7.5 Muscle and bone rigging system used to animate *Shrek*. Figure 5.7.4 shows the clothed skin model that was attached to this hierarchical system. (*Shrek*™ and © 2001 DreamWorks L.L.C.)



SKELETON AND AUTOMATIC SKIN



SKELETON AND MANUAL SKIN

SHREK, MANUAL SKIN

5.7.6 Skeletons with hierarchy information, with skins created automatically and manually. Skins modeled manually have more detail than those generated automatically. (*Shrek*™ and © 2001 DreamWorks L.L.C.)

tices. Some areas of the surface or specific elements in the model may be defined as **rigid geometry** that behaves like hard shells around the skeleton and does not deform automatically with rotations of the joints.

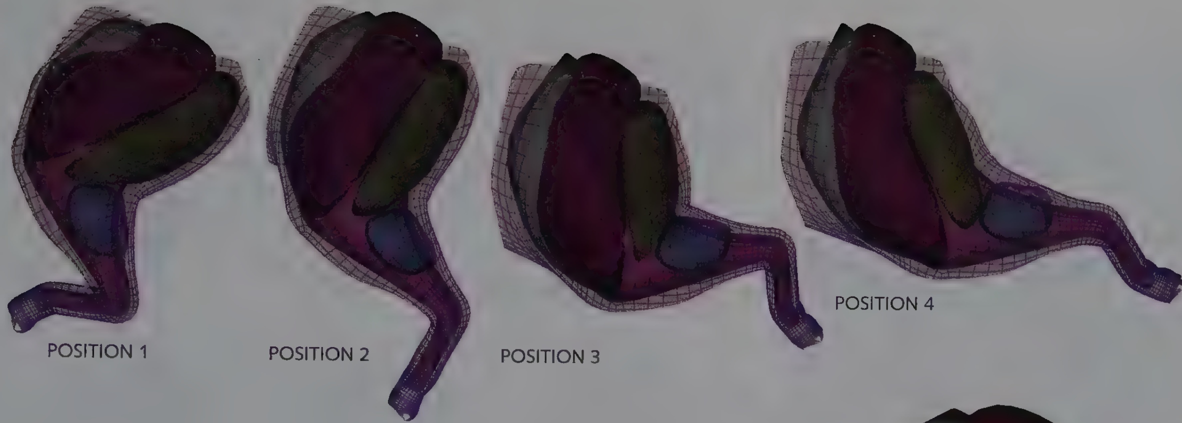
Many software programs provide utilities tools for the automatic generation of rigs based on a particular surface and its topology (Fig. 5.7.10). and automatic binding. Some software also offers the possibility of generating a skin-like surface that wraps the skeleton or articulated chain. Skin surfaces can be generated automatically by creating an outline shape and letting the software extrude it along the skeleton (Fig. 5.7.6). In most cases, the surface that results from **automatic skin generation** can be edited manually.

In some cases, the skin of three-dimensional models is not defined as a **continuous surface** but instead as a series of independent surfaces that are related to each other through a hierarchical structure. In these instances, and depending on the desired effect and style, the surfaces representing the skin should be arranged with care to minimize the number of **gaps on the surface** of the animated model. These gaps are not an issue if the figure is designed in such a way that the surfaces are not connected—for example, a cartoon character with unconnected floating body parts. But in some instances gaps on the skin surface tend to look like modeling mistakes. Two simple techniques that can be used to minimize the gaps on the skin of objects built with several surfaces are: using filler shapes at the joints, and plain-shape interpolation. **Filler objects** are usually spherical and positioned at the joints that have wide rotation angles, which is where gaps in butting surfaces are most likely to occur. Simple **shape interpolation** can also be used to patch small gaps that may develop during animation on models with multiple surfaces. But results are often far from perfect because of the unexpected skin distortions generated when the interpolations are applied to gaps created with elements rotated around multiple axes.

Skin Deformers

The animation rig controls and deforms the skin geometry of the character in a variety of ways: with simple joint attachments, using the actual bone geometry, or a complete muscle system. Binding skin points to one or several joints is the simplest technique for skin deformation. In order to increase the complexity of their translations these skin points can be weighted, or influenced by the rotations of multiple joints. In addition to straight-forward weights and constraints, a variety of geometry deformers can be included in a rig. A few of these include, for example, commonly-used blend shape deformers, cluster deformers, lattice deformers, jiggle deformers, and wrinkle deformers.

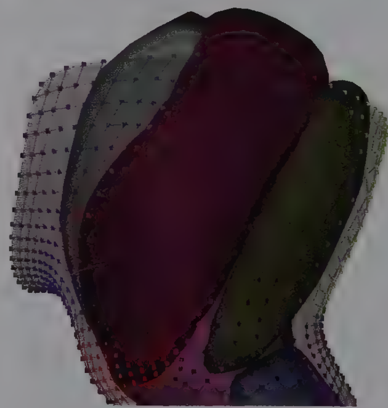
The skeleton geometry itself (not the abstract hierarchical skeleton) can also be used as a skin deformer. In this case the bone geometry is parented to the inverse kinematics (IK) joints and used as skin weights. Using bone geometry as a skin deformer serves the dual

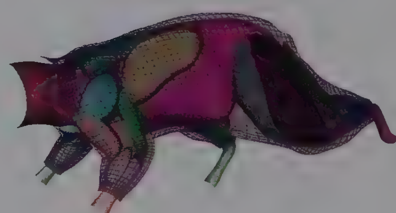


purpose of being a useful visual aid for positioning skin, joints, and muscles in the proper place. The bone geometry is only used to extract weight values and "bake" them into the skin, after which point the skeleton geometry can be removed. Baking can be done by assigning the weight information to the skin without the bone geometry, saving the skin weights to a file, removing the skeleton, and reassigning those values to the skin without the bones.

A more sophisticated technique for skin deformation consists of building geometry that simulates muscles. The geometry of these virtual muscles is usually designed with several control points so that it maintains its volume as it flexes. The muscles are usually connected to the skeleton by following its hierarchy: the start of the muscle is connected to parent bone and the end to the child. Collision detection is used to set up the muscles so that they slide along or bulge as they press against a bone or other muscles (Figs. 5.7.5 and 5.7.8).

Muscle rigs can have different degrees of realistic motion depending on the complexity and functionality of the geometry used to simulate them. Muscles built from a single piece of geometry can be set up to twist along with a bone twist. Spheres are useful simple shapes for modeling a large variety of muscles, but groups of muscles often need to be modeled with more complex shapes (Fig. 4.5.1). Several geometry strands can be used to achieve more detailed motion of the muscles and the skin attached to their surface. Multiple curves and lattice deformers can also be used to control more subtle deformations of the muscle surface. Lattices are easy to set up (Fig. 4.5.5) but can be less precise than a carefully constructed cluster of control curves.





5.7.8 The muscle system that drives the shape of the skin is revealed in these four steps in the bending of a quadruped's hind leg. The skin morphs as the muscles swell or contract. The colored dots indicate the different areas of the skin that are bound to a particular color-coded muscle. (Images courtesy of Hans Rijpkema, Rhythm & Hues Studios.)

5.8 Getting Ready

Overlapping Edges and Gaps

Many three-dimensional software programs have trouble rendering objects with **overlapping edges** or overlapping facets. Clean up your three-dimensional models before you pass them over to the team

5.7.9 Model showing the muscles and the rigging controls (top), and the skin surface (bottom). (Teenage Mutant Ninja Turtles and TMNT are trademarks and copyrights of Mirage Studios, Inc. *TMNT* © 2007 Imagi Production Limited.)

members in charge of rendering them or before you render them. Use modeling utilities such as aligning or blending to eliminate overlapping objects. In critical situations automatic aligning tools may not be available to fix the misalignment of two elements and the resulting gaps on the rendered surface. In those cases it is often preferable to align the objects by typing their exact XYZ position in space instead of trying to drag them into position with the mouse.

Disable Polygons that Face Away

When building a large model that is to be rendered only from a single fixed point of view—front view only, for example—it is wise not to model the **surfaces facing away** from the camera and to disable the **back faces**. This technique is reminiscent of stage sets that are perfectly finished on the sides that face the audience but are raw and unpolished on the sides that cannot be seen from the theater. Some programs allow users to turn off the rendering of individual polygons or groups of them. Other programs require users to slice off the unwanted surfaces and discard them altogether from the three-dimensional scene. Either technique is useful for single-point of view renderings because it yields simpler models, more compact file sizes, and faster renderings.

Keep Heavy Motion Rigs Separately

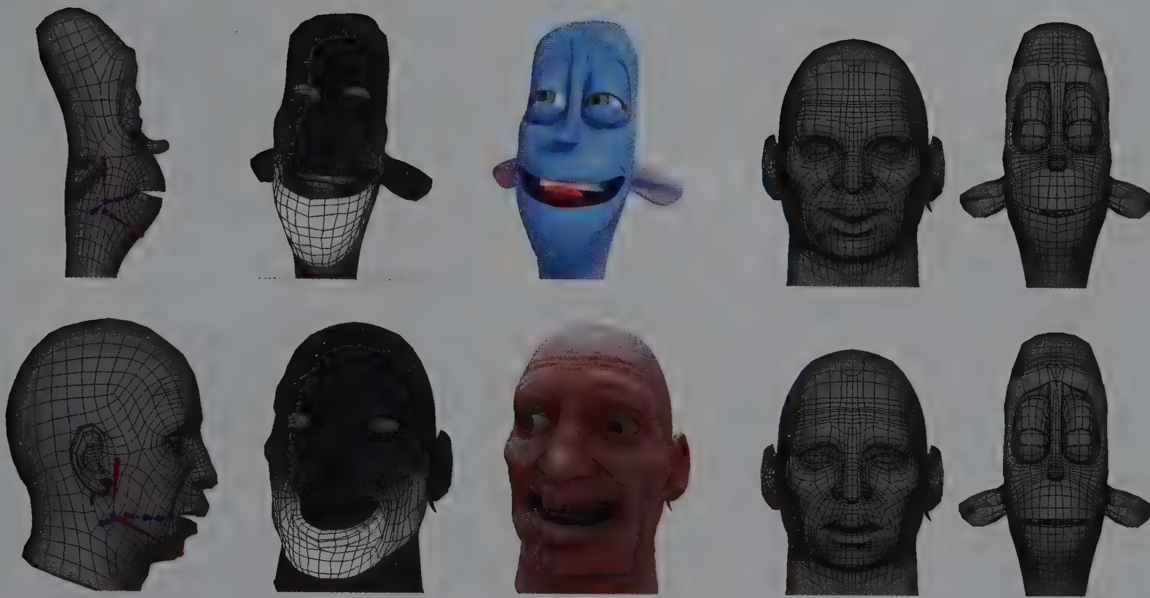
Complex rigs can become cumbersome and may require significant computing to apply all the controls to the skin. For this reason "heavy" motion rigs may be kept as separate components or layers. This offers more flexibility and allows animators to start animating with a lighter rig while the technical director refines the skin mesh and the muscle deformers of the final rig.

Stay Away from Overmodeling

Creating too many elements, or **overmodeling**, is a bad habit that usually has negative consequences in the stages of rendering and animation. Streamline the size of your models and data files by keeping the number of polygons, curves, or points on the curves down to an absolute minimum. If necessary, use purging or blending utilities or edit your models manually. The numerous complications created by overmodeling are never obvious during the modeling process, but they become painfully clear later, during animation and rendering.

Take Advantage of Modeling Mistakes

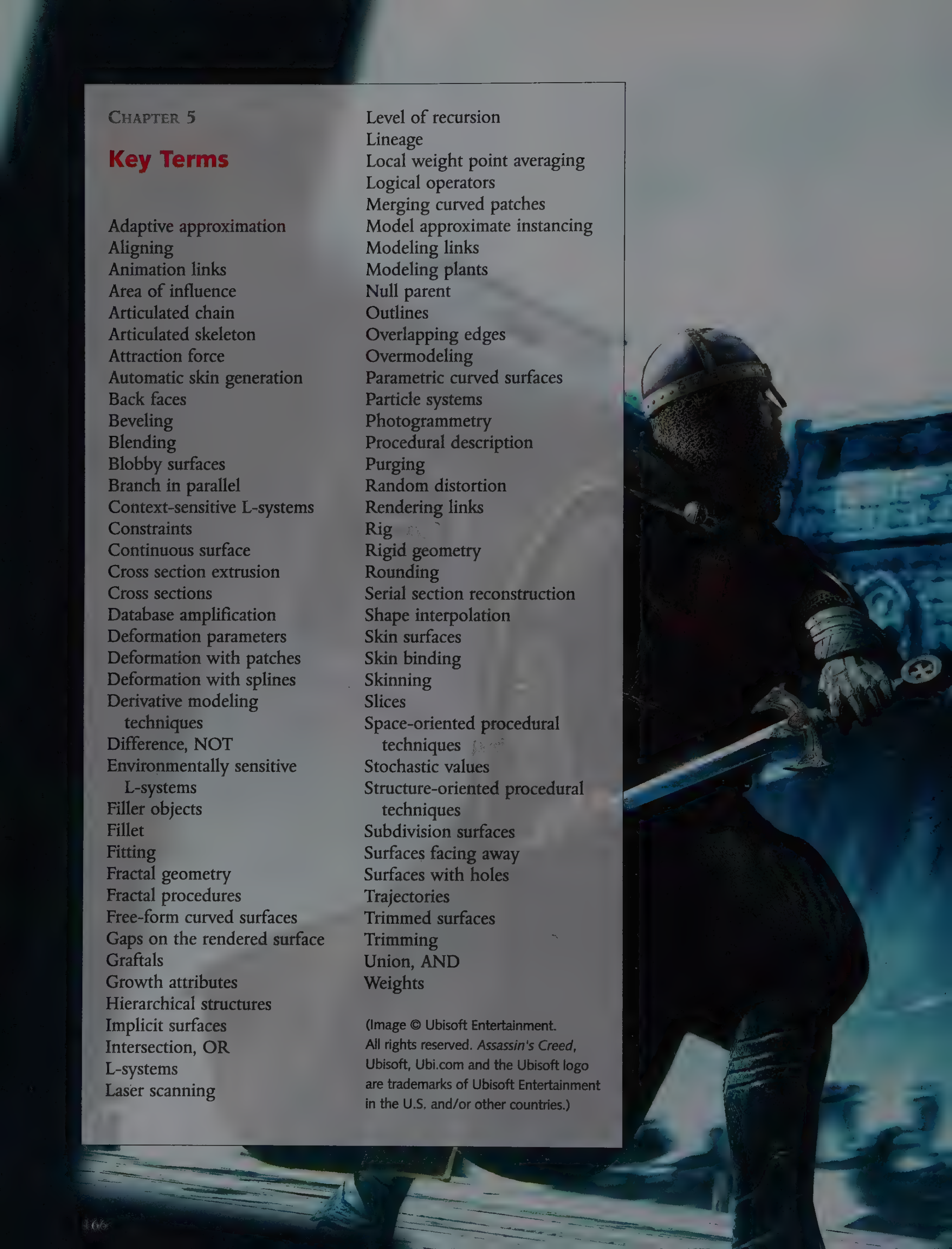
During the modeling process, one often encounters tools or functions that do not quite behave the way they are supposed to. This is particularly true of **derivative modeling techniques**, such as skinning and logical operators, that create objects from other existing objects.



The limitations of these modeling tools is apparent when one tries to build unusual surfaces such as zig-zagged shapes, overlapping concave areas, or a large number of acute angles. Modeling mistakes or accidents also happen when we try to get the tool to do something that it was not intended to do. For example, asking a skinning tool to first connect the two contours on opposite edges of the object and then proceed inward until all the contours are connected in this fashion instead of just skinning the adjacent contours serially and in sequence from one side of the object to the other. While the malfunctions of modeling tools are disappointing at first—like any other defect or "bug" in a program—we can learn to look at them in a fresh way and seriously consider whether the results can be used in a creative way to enrich the scene. Think of the ways in which the painters of the Abstract Expressionist movement of the 1950s used the shapes of accidental paint dripping and the unpredictable patterns of energetic brushstrokes to build their marvelous works.

Use the full potential of virtual modeling by exploring shapes, objects and motions that do not exist in reality or that would be impossible to build in reality (Page 1 and Fig. 4.7.2). Combining direct modeling with procedural methods can yield particularly interesting results.

5.7.10 Facial expressions can be transferred between characters by using readjustable rigs. (Left, top and bottom) A laughing pose as it transferred between a human and a cartoon character. The jaw joints are shown in red, the tongue joints in blue, and the transfer weights as a monochromatic map. (Above) Side-by-side comparison of a sad brow and a smiling mouth. (© 2007 Verónica Orvalho, www.faceinmotion.com.)



Key Terms

Adaptive approximation
Aligning
Animation links
Area of influence
Articulated chain
Articulated skeleton
Attraction force
Automatic skin generation
Back faces
Beveling
Blending
Blobby surfaces
Branch in parallel
Context-sensitive L-systems
Constraints
Continuous surface
Cross section extrusion
Cross sections
Database amplification
Deformation parameters
Deformation with patches
Deformation with splines
Derivative modeling techniques
Difference, NOT
Environmentally sensitive L-systems
Filler objects
Fillet
Fitting
Fractal geometry
Fractal procedures
Free-form curved surfaces
Gaps on the rendered surface
Graftals
Growth attributes
Hierarchical structures
Implicit surfaces
Intersection, OR
L-systems
Laser scanning
Level of recursion
Lineage
Local weight point averaging
Logical operators
Merging curved patches
Model approximate instancing
Modeling links
Modeling plants
Null parent
Outlines
Overlapping edges
Overmodeling
Parametric curved surfaces
Particle systems
Photogrammetry
Procedural description
Purging
Random distortion
Rendering links
Rig
Rigid geometry
Rounding
Serial section reconstruction
Shape interpolation
Skin surfaces
Skin binding
Skinning
Slices
Space-oriented procedural techniques
Stochastic values
Structure-oriented procedural techniques
Subdivision surfaces
Surfaces facing away
Surfaces with holes
Trajectories
Trimmed surfaces
Trimming
Union, AND
Weights

SECTION III

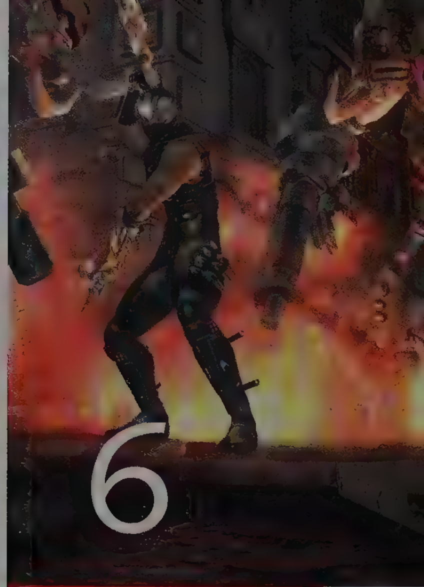
Rendering

(Opposite page: *Ninja Gaiden*™, Xbox real-time render. Reprinted with permission from Microsoft Corporation.)

6.1.1 The masterful use of lights, cameras, and materials makes it difficult to decide what in this picture is real and what is computer generated. Tries to capture the essence of the city of Vancouver in a quiet afternoon before the ski season starts. (Image courtesy of Aketoshi Tada, ataKikaku Co., Ltd, and Next Limit.)

CHAPTER 6

Rendering Concepts



Summary

Most of the visual characteristics of a simulated three-dimensional environment are determined during the rendering process. This chapter provides an overview of color concepts, and a variety of rendering methods, including ray tracing, radiosity, hardware-based, image-based, and non-photorealistic rendering.

QUICK INDEX

6.1 Lights, Camera, and Materials

The world of rendering three-dimensional scenes and characters with computer software is populated by most of the attributes of our visual realm where the shapes of objects are revealed by light and obscured by shadow, where color creates moods of subtle tranquility or explosive happiness, where textures are as delicate and lyrical as fine sand or as dramatic and forceful as malachite, where the restless translucency of rain distorts the features of the world and the mirror of transparent water puts them back together. Rendering worlds of reality or fantasy with the computer can create results as artistic as with any other medium (Fig. 6.1.1).

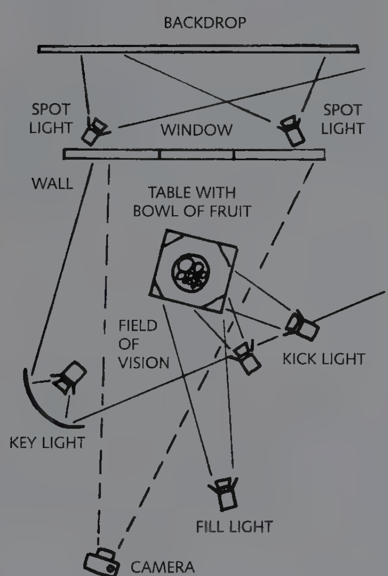
When we use computers to render real or imagined scenes we can follow specific procedures, as we would if working with other media, that help accomplish all the tasks required before the rendering can be completed by the computer program. This fact is understood by creators who use media like paint, photography, or cinematography to depict scenes, express emotions, or tell stories with the elements of visual language. Each group of image-making professionals has developed ways of doing things—basic techniques, orders of execution, and even complex procedures to do their jobs. Most visual artists have to deal with a few basic elements during the image-making or rendering process. These elements include composition, lighting, and defining surface characteristics such as color and texture.

Let's, for example, consider all the preparation that takes place before a professional photographer is ready to push the camera's

trigger and actually take a photograph. The tasks that the photographer must complete will lead to the creation of an image. Likewise, when using three-dimensional computer software to simulate a realistic or imaginary scene, many tasks and variables have to be taken care of before the program can process the information needed to generate the image we have in mind. The tasks needed when rendering by computer are similar in theory to the tasks of the photographer. But in practical detail, the tasks are carried out with different tools and in a different environment.

The photographer in our example is photographing an arrangement of fruit in a bowl placed on a small table in the middle of a large room with a bright landscape showing through the only window in the room (Fig. 6.1.2). The photographer has to start by selecting the models, or "props," themselves. That includes the fruit, the bowl, the small table, and the tablecloth. The props may be procured from a variety of places that might include a farmer's market, a housewares store, and an antiques shop. The objects and the fruit have to be chosen according to a certain criterion that specifies the way they are supposed to look. The table and the tablecloth, for example, might have to have a certain soft antique look. The bowl, on the other hand, might have to be made of dark translucent crystal and have an elegant and simple design. Only the freshest fruit can be used for this shot. Once all the objects have been brought to the studio, then the photographer and her assistant will arrange the objects in a specific composition.

6.1.2 The still life is arranged close to the fake wall in the photo studio so that the window and the painted backdrop showing through it appear behind the table. The top view of the shoot location is below.



In three-dimensional computer graphics all of the objects used in the scene are simply called **models**. We learned in Section II of this book that three-dimensional models can be built with a variety of software techniques. Once the models have been built, they can all be placed in the **virtual studio** that exists in the computer's memory and arranged in a specific way by using a combination of geometric transformations.

Let's return to the photographer, who is in the process of placing the table near the panel in the middle of the room and the camera in front of it so that the window on the panel appears behind the table. She also arranges the fruit in the bowl. She steps back, looks at the fruit through a professional photographic camera that is mounted on a tripod that is close to the subject, and returns to rearrange the fruit. At this point the photographer takes a snapshot with a hand-held camera loaded with instant film and a flash, just to get a quick preview of the composition. Because of the flash of intense light emitted by the flash lamp, the color in some areas of the instant photograph washes out, the overall illusion of depth is somewhat flattened, and the delicacy of some textures is lost. But the instant flash photograph does the job: it records the position of objects in space and the composition. It is also possible to create such quick snapshots with three-dimensional rendering software.

All the components of the still life are now in place, and it is

time to start playing with the position, intensity, color, and focus of the lights. Since the shot is taking place inside the studio, very little natural sunlight is available. Therefore, the photographer must recreate not only the sunlight but also some lights that must be focused on several areas of the fruit in order to delineate the shapes with more clarity and to accentuate the highlights and the shadows on the fruit to further the effect of depth.

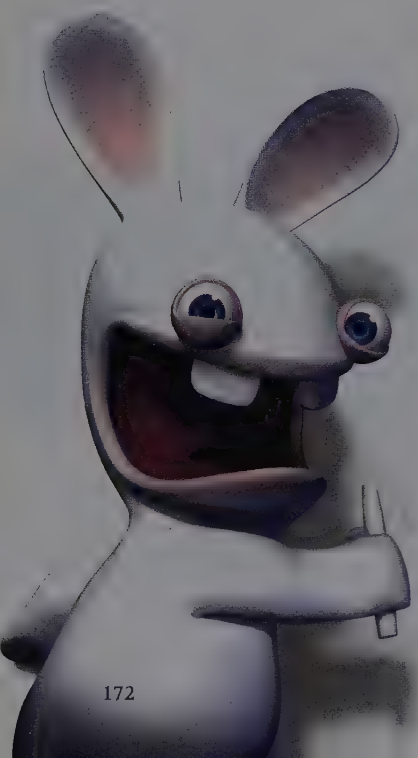
Our imaginary photographer pauses for a minute, looks around the almost empty and dark room, and tries to visualize the effect of different types of lights on the fruit, the tablecloth, and the walls. After discussing some of the stylistic possibilities and production implications with her assistant, the photographer decides to start with one intense but indirect flood light for simulating the effect of warm natural light of a medium intensity. The flood light is placed between the still life and the camera, but away from the still life and into a concave portable reflective surface in the shape of an umbrella.

Then the photographer carefully maneuvers three small spotlights pointed at different fruits in the bowl. Since the three spotlights all have the same intensity, the photographer has to move some of them closer or farther away from the fruit depending on the lighting effect desired. While the photographer is still arranging the three spotlights, her assistant is busy lighting the backdrop (with a landscape painted on it) that is placed behind the fake wall and visible through its window. The photographer's assistant decides to use two small flood lights with a slight blue coloration in order to simulate the exterior light of a cold rainy day. The two small floodlights are pointed at 45-degree angles from the back of the fake wall directly onto the backdrop (Fig. 6.1.2). Most three-dimensional rendering programs allow

6.1.3 The haunting lighting and rendering effects in *The Cathedral* were rendered with the Brazil ray tracing renderer. (Copyright 2002 Tomek Baginski and Platige Image.)

6.1.4 Uniform ambient lighting and bright overhead lights were used to light this low angle shot, from the *Free Jimmy* animated feature. (© 2003 AnimagicNet Norway.)

6.2.1 A CMYK full-color image is created when the four image layers are viewed together. Four-color separations (cyan, yellow, magenta, and black) are employed in most mechanical reproductions of color images in paper-based magazines and books.



artists to select and place light sources with the same amount of intuitive trial-and-error and precision as lighting with real lights.

During the placement of the lights, the photographer and her assistant go back and forth to the camera to check through the viewfinder whether the lights are defining the image composition and mood that they had in mind at the onset of the process. At first, double-checking the lights is intuitive and purely visual. But before taking the final photograph it becomes necessary for her to measure light in a systematic and precise way. This measurement is done with a special device called a **light meter**, which provides the photographer with a precise, numerical value that represents different characteristics of the incident light at any point in the three-dimensional environment.

It is also necessary, as part of the light measuring process, to double-check and adjust the light readings against other numerical values involved in the process of photographing the still life. Those other numerical values might include, for example, the speed and chromatic characteristics of the photographic film, the chromatic value of the filter placed on the lens of the camera, and the chromatic value of the reflected light. This constant back-and-forth double-checking is not only necessary to stay on the desired track but is also an integral part of the creative lighting process.

During the visual checking done by looking through the viewfinder, the photographer's assistant notices that the surface of one of the fruits looks somewhat dull and slightly flat, and that is not the way fruit is supposed to be portrayed in this image. The photographer and her assistant determine that this might be because the skin of the fruit became too dry while in storage. They also agree that the best way to fix this—short of getting a replacement fruit—would be to apply a thin coat of oil to the skin of the fruit. Likewise, the surface characteristics of all objects can be easily determined and fine-tuned with most three-dimensional rendering software.

Once all the lights and surfaces have been fine-tuned it is time to make slight adjustments to the placement and focusing of the camera. The photographer decides to replace the lens of the camera with another one that provides a wider field of vision so that a larger portion of the scene will fit in the final image without having to move the camera farther away (Figs. 7.4.3 and 7.4.4). Finally, the scene is ready to be recorded. Three-dimensional rendering software provides tools for simulating lenses of different focal lengths as well as the effects, such as depth of field, associated with them (Figs. 4.6.3, 6.1.3, 6.1.4, and 7.2.3–7.2.10).

6.2 Color Models

This section covers some of the most popular color models used in three-dimensional computer rendering including Cyan, Magenta, Yellow, and Black (CMYK), Red, Green, and Blue (RGB), and Hue, Saturation, and Lightness (HSL). See Chapter 13 for more information on color resolution and color look-up tables.

Additive and Subtractive Systems

Those of us who learned about color theory as kids were oftentimes given three jars each with red, blue, and yellow paint, and were asked to create new colors by mixing the paint from each of the three jars. I was told when I was five years old that the three primary colors are red, blue, and yellow (RBY). But that was only part of the story. Those three colors are indeed primary colors but only in a paint-oriented **pigment-based color** environment. The RBY **subtractive color system** is useful for understanding color relationships in a paper-and-paint environment.

Cyan, Magenta, Yellow, and Black

Another model that is used to explain and define colors that are pigment-based is the **Cyan**, **Magenta**, **Yellow**, and **Black (CMYK)** color model. This model is widely used in traditional graphic arts and digital printing for mechanical reproduction of color images (Fig. 6.2.1).

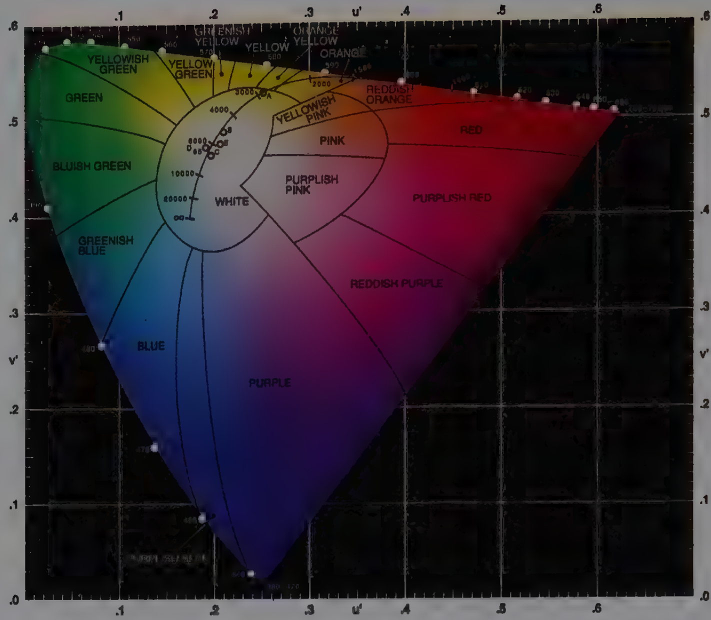
Red, Green, and Blue

The colors displayed on a computer screen, however, occur in a **light-based color** environment. These colors are created by combining different amounts of the three primary colors in the **additive color system**: **Red**, **Green**, and **Blue (RGB)**. In the RGB model colors are defined in terms of their amounts of red, green, and blue. Combining—or adding—all the primary colors in the RGB system yields white, therefore the name additive system. Combinations of two primary RGB colors yield results that would not make sense in a pigment-based model. For example, the combination of red and green lights in equal proportions yields yellow light. The RGB color model takes advantage of the technology used in computer monitors. It is a precise and efficient way of describing color, and it is used in all computer systems. Most software represents each of the RGB primary colors in a separate layer or channel so that each color can be manipulated independently (Figs. 6.2.2 and 6.2.4). RGB colors can be described by specifying numerical values that may range, for example, from 0 to 1, or from 0 to 255. The value of pure green is 0-255-0. The value of a greenish light blue could be 150-200-255, and a yellowish dark orange 120-80-30 (Fig. 8.3.2).

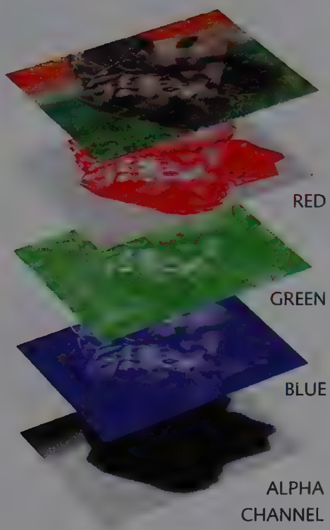
Hue, Saturation, and Lightness

Working with the RGB model can be confusing at first. The **Hue, Saturation**, and **Lightness (HSL)** is a more intuitive alternative color model that can be used for specifying color in a light-based environment. In the HSL model, colors are described in terms of their hue, saturation, and lightness. The HSL color model can be

6.2.2 These beautiful patterns display the values of color samples in the RGB, HSB, and CMYK color models. The RGB values range from 0 to 255, CMYK ranges from 0 to 100%, and HSB is expressed in degrees (hue) and percentages (saturation and brightness). (Courtesy of Akira Kai, FOTON).



6.2.3 This diagram of the CIE color space defines the colors in the electromagnetic spectrum. Those visible to the human eye are inside the inverted D-like shape, those beyond the range of visible color—like ultraviolet and infrared color frequencies—are outside of the shape.



6.2.4 Each of the RGB primary colors is kept in a separate layer or channel. These image layers can also contain information other than color. An alpha channel, for example, contains a black-and-white image that can be used as a stencil to mask some areas of the layers being composited with other images. (Related masks can be seen in Figure 14.1.9.)

visualized as a three-dimensional space that simplifies the location and description of color. This space can be defined by two cones connected at their base. The vertical axis that runs between the two peaks is used to define the **lightness**, or darkness, of a particular color. The lightness of a color increases or decreases along the direction of this axis, with the lighter colors located at the top of this space, and the darker ones at the bottom. The **saturation** of a color increases along a line that starts perpendicular to the vertical axis of the space and ends at the outside surface. The colors on the outside surface are fully saturated, while the colors close to the center of the space are washed out. The **hue** of colors in the HSL color space changes with the angular position of the color around the vertical axis. All the spectrum of hues can be found around the vertical axis, and their position can be specified in degrees.

Slicing the HSL space horizontally reveals many hues and saturation values with the same lightness value. The gray colors—neutral and colorless—are found on the vertical axis. Absolute white and absolute black can be found in the peaks of the HSL space. A popular variation of the HSL color model is the **Hue**, **Saturation**, and **Brightness (HSB)** color model. The main difference between the HSL and the HSB models is that the latter uses two cylinders—instead of two cones—to define the chromatic space. A graphical representation of a section of the HSB color space is shown in Figure 8.3.4.

Chromatic Range and Color Space

It turns out that not all media and techniques we use for creating color are capable of creating exactly the same colors. Each medium

has its own distinct **chromatic range**. This explains why so often the colors we see on the screen are different from the colors we see on the printout. A common problem one encounters when creating computer images for output on videotape, for example, is that some colors created with RGB computer monitors are too bright and saturated for display on standard television sets. Those working in RGB for eventual output to video have to clip the RGB colors that are outside the video color range in order to avoid distortions, such as color bleeding, from showing in the final video recording. This is illustrated in Figures 6.2.5 and 6.2.6 where a color sample of the RGB color model is translated into the NTSC and CMYK models. See Chapter 15 for additional information on NTSC.

Knowing which color ranges overlap between a variety of media and formats, and which do not, is helpful in devising solutions to work around the physical limitations of color reproduction. The **CIE color space** is a useful aid for visualizing the chromatic ranges of different media. In the early 1930s the International Commission on Color, known as CIE (*Commission Internationale de l'Eclairage*), first presented a color space that defined all the colors in the spectrum that are visible to the human eye—ultraviolet and infrared color frequencies, for example, are beyond the range of visible color (Fig. 6.2.3).

6.2.5 The NTSC color space lacks the chromatic range to display some of the bright yellows and blues that can be created in the RGB color space. The light areas in the lower circle represent those colors that get clipped when translating a color from RGB to NTSC.

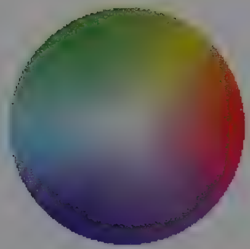
Conversions Between Color Models

When creating images with computers, it is often necessary to convert from one color space to another. This is mostly due to the fact that the creative process usually takes place in both light-based and pigment-based color environments. When we scan photographs or drawings we deal with pigment-based color. When we manipulate existing images or create new images on the screen we deal with light-based color. When we print one of our images on paper we deal, again, with pigment-based color.

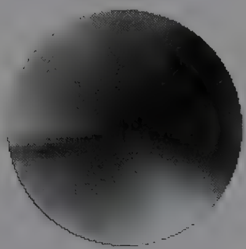
In most cases this **color conversion** is done automatically by the computer software that we use. Obviously, the quality of these color conversions varies from program to program. If we are pleased with the color output of our computer system in general, we do not have to get involved in the color conversion process. If we are not satisfied with the automatic or default color conversion, however, it is important to get involved in the color balancing and correction or enlist the help of a color-experienced user.

6.3 Steps in the Rendering Process

The overall rendering process consists of a few basic steps regardless of the software used. It is not required that these steps happen in a rigid order. In fact, some projects might require that the sequence of steps is slightly altered. Additional details on the entire production process can be found in Chapter 2. Because of the complex and cumulative nature of the rendering process there is usually consider-



RGB AFTER CMYK CLIPPING



RGB/CMYK DIFFERENCE

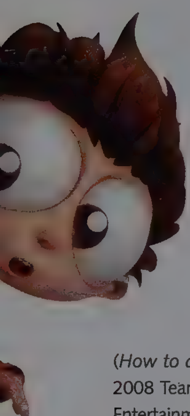
6.2.6 Many of the bright, saturated colors typical of the RGB color space cannot be reproduced in the CMYK chromatic range. The color circle shows a section of the RGB colors that were not clipped when translated to the CMYK inks on this page. The light areas in the lower circle represent the colors in that section of the RGB color spaces that were clipped off the CMYK gamut.

able bouncing back and forth between stages before the process is completed. But keep in mind that the implementation of general techniques in different programs and in different production studios might require slight variations in the sequence of steps described here. Figure 6.3.1 summarizes the main steps in the **rendering process** in the form of a flowchart. Each of the steps in the rendering process is covered in detail throughout Section III of this book.

The first step in the rendering process starts with well-built and fairly complete models—characters, sets, and/or props—to be rendered, including their UV maps. We also determine at this point if texture maps are required and if they have already been produced. Secondly, we make sure that the rendering pipeline is clear based on expectations, and that it is also understood by everyone who we will need assets from or give assets to. We maneuver the camera in XYZ space so that we find a portion of the environment or geometry that we want to focus on while working. We might reposition the camera, tilt it, change the focal point and the depth of field, and adjust the proportions and the resolution of the image. If necessary, we may further rearrange the objects in the scene. Third, we design and implement the lighting scheme according to the general lighting indications. We run a few quick tests to check the results and show to others for feedback and/or approval.

Next, we specify the characteristics of the surfaces of the objects including color, texture, shininess, reflectivity, and transparency. Custom shaders might need to be developed. This often requires a fair amount of attention to detail, and doing a good job during this stage will have a great impact on the quality, refinement, and energy of the final rendering result. Once all the basic ingredients are ready and "in the pot," we stir and cook slowly. Finally, we run shading tests and adjust parameters, most likely ending up with a multi-layer multi-pass render that we can submit for feedback and approval.

When the geometry or shading in any given scene are too complex it is common to render different components of the scene separately. This is usually called working or **rendering in layers**. A simple example of this method would consist of rendering the background by itself, then rendering the foreground elements, and finally compositing them together using the techniques described in Chapter 14 (Fig. 14.1.1).



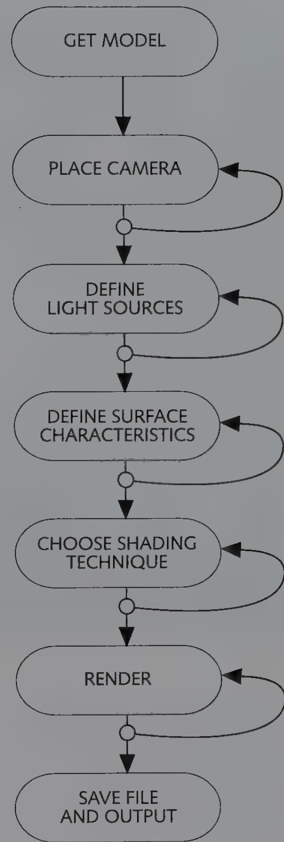
(*How to drive everybody crazy*, © 2008 TeamTO–France 3–Cake Entertainment.)

Rendering Methods and Compositing

A large variety of rendering methods can be used to turn a wireframe view of a three-dimensional model into a shaded image. In addition to the skilled placement of light sources in a scene and the assignment of surface characteristics to objects, the look and quality of the final shaded image is largely dependent on the type of rendering and compositing methods used. Some rendering programs are supplied as "black boxes" that only accept geometry data and rendering variables—such as lighting, shading, and surface characteris-

tics—and cannot be modified by the user. But there are a few rendering environments that offer users the ability to get under the hood and write custom shaders, among other forms of tweaking. Some rendering programs also provide extensive technical notes explaining how they work, and such notes can provide useful insights about the ways in which the program renders. Knowing this information can prove invaluable when setting up a scene for rendering. But in most cases it takes some practice to get a feel for or to understand how a specific shader or shading program renders. Knowing the strengths and weaknesses of a rendering program can be advantageous for many stages of the three-dimensional creative process with computers, ranging from the sketching and storyboarding stage all the way through final output and recording.

You should keep a couple of practical ideas in mind about rendering in general. Each of the rendering methods described in this section has particular strengths and weaknesses. Often the same rendering method is implemented slightly differently in different software programs, so the results of using the same rendering method with the same variables in two different software programs cannot be expected to be the same. Finally, keep in mind that several rendering methods can be used in conjunction with one another. This is usually called **hybrid rendering**. Figures 6.7.4 and 6.12.4 show the result of rendering a scene in layers and with both ray tracing and radiosity rendering techniques. Another thing to keep in mind regarding the rendering process has to do with compositing. Long gone are the days when rendering was finished as soon as the render job was done. The trend today is to render a scene in layers, combine them in compositing, rerender a couple of layers, and assemble the final render in compositing (Figs. 14.1.4, 14.1.10, and 14.5.1).

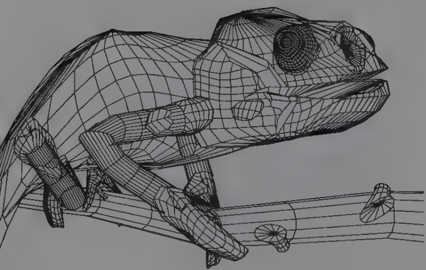


6.3.1 The condensed steps of a fairly standard rendering process. After some of the steps a test is made to see if the results are on track. If not, the variables are adjusted. See the flowcharts in Chapter 2 for overall digital production pipelines that are fine-tuned to a specific type of project, team configuration, budget, or deadline.

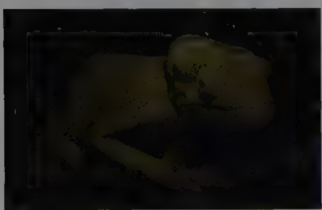
6.4 Hidden Surface Removal

The hidden lines and surfaces of the object that are not visible from the point of view of the camera must be removed before the three-dimensional geometry can be rendered. Several algorithms have been developed to sort all the points, lines, and surfaces of an object and decide which of them are visible and which are not. Then the **visible surfaces** are kept and the **hidden surfaces** are removed (Fig. 6.4.1). Before the polygonal meshes and free-form curved surfaces can be rendered, they are subdivided into triangles; this process is called **tessellation**. As explained in Chapter 9, the removal of hidden surfaces is determined by the relationship between the orientation of their surface normals and the position and orientation of the camera. There are many methods for hidden surface removal, and they can be separated into the general categories of object space and image space. A few hybrid rendering methods operate in both object and image spaces.

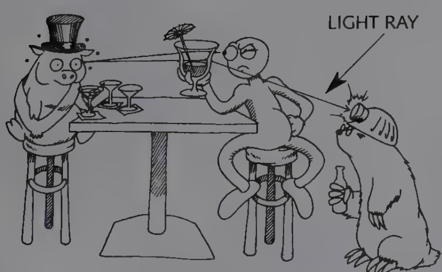
Object space methods for hidden surface removal make the calculations in three dimensions. These algorithms require intensive



6.4.1 A chameleon modeled with curved surfaces and then converted to a polygonal mesh. Rendered in wire-frame mode and with the hidden lines removed. (Courtesy of Tim Cheung.)



6.5.1 The color on the visible surfaces of this chameleon was calculated at each pixel. The total Reflected Light that bounces off the surface of the model equals the product of the Incident Light multiplied by the Surface Reflectivity.



6.6.1 The ray tracing process follows the path of light rays that reach the camera back to their light source.

computing and are unique because they generate data that can be used to improve the rendering of textures, shadows, and antialiasing. Ray tracing is an example of object space rendering methods, and it is the standard rendering method of many of today's rendering programs. Ray tracing follows the rays of light—emitted by the light sources in the scene—as they bounce off or travel through objects in the scene and eventually reach the camera. Object space hidden surface removal methods are analogous to global rendering methods because both take into account the entire scene information, not just a local single object, in order to perform their calculations.

Image space methods for hidden surface removal retain the depth information of the objects in the scene, but sort from a lateral position, and only to the resolution of the display device. Image space rendering methods render a three-dimensional scene by projecting the models onto the two-dimensional image plane (Fig. 6.4.1). Image space algorithms are generally efficient but discard some of the original three-dimensional information that can be used for shadowing, texturing, and antialiasing enhancement. Many of the image space methods for the removal of hidden surfaces were first developed in the early 1970s, and include Warnock's **area subdivision** (1969), Watkins' **scan line** (1970), and Newell's **depth sort** (1972). The latter is generally considered a hybrid image/object space algorithm. Multiple improvements and refinements to each of the original image space rendering procedures have been made and continue to be made.

6.5 Z-Buffer

The **Z-Buffer** rendering method is a popular image space rendering technique that incorporates some aspects of object space rendering. Z-Buffer rendering gets its name from the fact that all objects in the scene are sorted by their Z position, or depth, in the scene. This depth information is kept in a buffer, and made available to the rendering process as the hidden surface removal calculations are performed.

The Z-Buffer rendering method makes the hidden surface removal one object and one pixel at a time. This method determines whether an object is visible from the point of view of the camera at each pixel, and one pixel at a time. If the object is visible, its depth information (or distance from the camera) is checked, and this determines whether the object is the closest one to the camera up to that point in the sorting process. If it is, the object is shaded at that pixel, and this visibility test is repeated for the same object at all pixels on the screen. When the visibility of one object is completed, another object is chosen, and the visibility process is performed at all pixels all over again (Fig. 6.5.1). It is easily determined whether a new object is closer to the camera than the object tested earlier by checking the depth information of the new object. If the new object is closest to the camera, its shading values replace the previous shading values at that pixel. The Z-Buffer rendering is completed when the

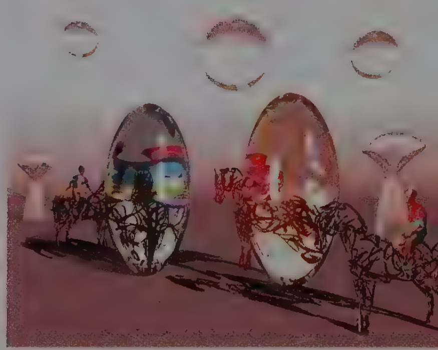
visibility of all objects in the scene has been tested in all the pixels. The depth information of a scene is also used in a variety of shading calculations that may be independent from the overall rendering technique. Depth information, for example, can be kept in a separate rendering layer commonly called a **Z-depth map**, or used to create depth-fading effects (Fig. 9.8.6).

6.6.2 A ten-level-deep ray-traced image with effects of reflection and refraction. The shattering was created with a dynamics simulation and baked into keyframes, with SoftimageIXSI and rendered with Mental Ray. See related figure on page 429. (Image courtesy of Kouhei Nakama.)

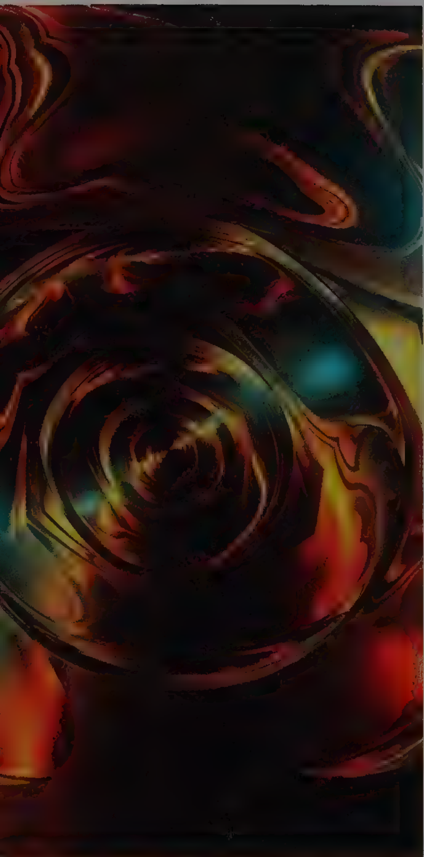
6.6 Ray Tracing

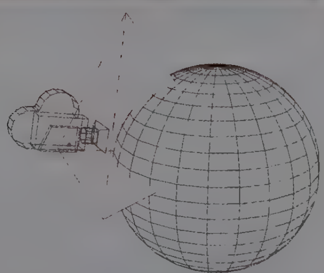
Ray tracing is a rendering technique capable of creating photorealistic images of three-dimensional scenes. Ray tracing is a sophisticated and accurate method for rendering, in part because it calculates every ray of light in the scene by following each one through the scene until it each reaches the camera. Ray tracing creates images with very accurate reflections and refractions of light, as well as detailed textures and shadows.

In general terms, **ray tracing** works by creating a ray for each pixel on the screen and tracing its path—one ray at a time—all the way back to the light source. A ray is an imaginary straight line that travels through three-dimensional space and collects rendering information. The values for the ray of light are calculated as it bounces off—or travels through—different surfaces in the scene with a variety of characteristics (Fig. 6.6.1). Ray tracing rendering techniques are based on the way in which rays of light travel from the light sources to our eyes, bouncing off surfaces that affect their characteristics along the way. However, it would be impractical to trace all the rays



6.6.3 A ray-traced image with effects of reflection and refraction of the ribbon-like structures used to define the horses. (Courtesy of Charles Csuri.)





6.6.4 Detail of the rich visual pattern created by the reflected intersections of a tetrahedron and a sphere. Modeled with Maya and ray-traced in Mental Ray. The camera setup for the rendering is also shown. See Figure 5.5.3 for a related image. (© 2008 Duncan Brinsmead.)

of light emitted by a light source in a scene because many of them never reach the camera. For that reason ray tracing programs trace the rays of light **backwards**, from the camera to the light source, to minimize the number of wasted calculations.

The main strength of the ray tracing rendering method comes from the fact that the image of a three-dimensional scene is calculated in three-dimensional space. The traced rays travel in three-dimensional space and often bounce from object to object. These rays are able to render reflections on shiny surfaces or refracted light in transparent objects. Unlike image space rendering methods, ray tracing is precise about simulating the behavior of light in three-dimensional space.

The primary controls in a ray tracing rendering are related to the depth of the ray tracing, the number of pixels in the image, and the number of light sources in the scene. The **ray tracing depth** is related to the number of times a ray will be allowed to come in contact with surfaces in the three-dimensional space (Fig. 6.6.2). Most subtle details in a ray-traced image are controlled by the depth of the ray tracing. Most ray tracing programs use different controls for the reflection rays, the transparency or refraction rays, and the shadow rays. Each of these types of rays calculate different components of the rendering of a three-dimensional scene. The depth of these three different types of rays is independent from one another. Figure 6.12.2 shows two versions of a ray-traced image based on different combinations of tracing depth. The first image contains a low reflection value and no refraction, therefore the transparent surfaces are not rendered as such. The second image contains a fair amount of reflections on the reflective surfaces, and refraction in the transparent surfaces, and simple shadows. It is important to keep in mind that when using the ray tracing rendering method, a surface that is made very reflective becomes like a mirror and, as a consequence, loses many of its other surface characteristics.

Reflection rays travel through the scene in a straight line, and they bounce off the reflective surfaces they hit at the same angle at which they hit them. Once a point in three-dimensional space has been hit by a ray and the value of that surface has been calculated, a **shadow ray** is shot from that point to the center of each light source in the scene. That point in three-dimensional space will only be visible if the shadow ray does not encounter another object before reaching the light source. When the ray tracing process encounters transparent surfaces in the scene, **refraction rays** are generated to calculate the amount of light refraction. In most ray tracing programs refraction is only enabled when the surface that is supposed to be refractive has a **thickness** defined by the distance between the front face and the back face of the surface (Figs. 6.6.3 and 6.6.4).

The number of rays traced in a scene—regardless of their tracing depth—is related to the spatial resolution of a scene, which is determined by the total **number of pixels** in an image. A single ray is traced backwards to a light source through every pixel in the image. Therefore, an increase in the number of pixels will result in an

increase in the number of rays traced and in the length of time that will be required to complete the rendering of an image. The **number of light sources** in a scene also influences the number of rays that are traced through the scene and, consequently, the length of the ray tracing rendering calculation. This is due to the fact that ray tracing works by tracing backwards each ray of light that reaches the camera.

Due to the intense computations required by the ray tracing rendering methods, many software programs provide simple ways to preview just a portion of the three-dimensional scene in the ray tracing mode. Some programs also estimate the length of time that will be required to complete a rendering and let the user know so that other tasks can be pursued in the meantime. In most cases the color of the light that is reflected by a surface is calculated as the combination of red, green, and blue light. The color of the reflected light is based on the color of the surface, the color of the incident light that reaches the surface, and on the reflectivity of the surface for each of the red, green, and blue components of light.

6.7 Global Illumination and Radiosity

6.7.1 In this radiosity rendering, the three-dimensional space is subdivided with an overlaid mesh that simplifies the calculations into a sequence of small clusters of source, receiver, and blocker polygons. (Image by Seth Teller, C. Fowler, T. Funkhouser, and P. Hanrahan. University of California Berkeley Walk-Through Group, and Princeton University Computer Graphics Lab. Courtesy of Seth Teller.)

Global illumination can create images that are more physically accurate than any other rendering technique because it calculates indirect illumination on objects, including diffuse, glossy, and specular **interreflection** between surfaces and transmission of light from other objects (Fig. 6.7.3). The global illumination rendering methods are based on the principles of illumination engineering theory and energy transfer. There are a few variations of global illumination rendering, including radiosity and photon maps.

Radiosity rendering focuses on diffuse interreflection between surfaces, and it typically divides the geometry in the environment into areas, or clusters, of polygons according to the way in which light affects them. The polygons in a radiosity calculation are typically catalogued into light sources, light-receiving surfaces, and light-blocking surfaces. By using iteration techniques, radiosity rendering calculates the amount of light transferred from one surface to another. This iteration or repetition is continued until the light energy is fully absorbed by the surfaces and/or it dissipates in space. With radiosity rendering, the subdivision of the three-dimensional space is based on the amount of light that is emitted or transferred between surfaces. The cataloguing of surfaces necessary to perform radiosity calculations generates data structures that look like **subdivision grids** when displayed on the screen (Fig. 6.7.1). These data structures typically require large amounts of main memory (RAM), and of raw computing. Once a grid of subdivisions has been established then the energy emitted by each light source can be followed throughout the environment based on the geometry of the surfaces. The distance between surfaces and their angular position are two important factors used in establishing the amount of light energy that can be transferred. Before light dissipates in space, much of it

6.7.2 Lighting simulation of the main lobby of the Eli Lilly Library at Indiana University Bloomington. One of the earliest images rendered with radiosity techniques, this image was created in the early 1990s with Greg Ward's Radiance version 2.2. The geometry was created with AutoCad Release 12 for the IBM RISC 6000. (Courtesy of Reuben Mcfarland and Scott Routen, Artifex, Bloomington, IN.)

6.7.3 (Facing page bottom) A detail of the XSI global illumination dialog box. (Courtesy of Softimage Co. All rights reserved.)

bounces off between surfaces if they are parallel to each other. But less energy is transferred if the surfaces are perpendicular, and none is transferred if they face away from each other. Likewise, more energy is transferred between surfaces if they are close to one another than if they are farther apart from each other (Fig. 6.7.2). One of the most striking lighting effects achieved with radiosity is **color bleeding**, which happens when colored objects pass some of their color through diffuse light to the neighboring objects. When rendered with radiosity a bright red sphere, for example, will cast a pale red diffuse interreflection on an adjacent white wall.

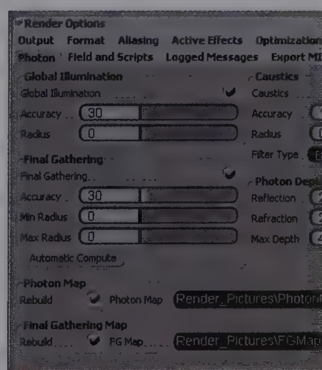
Radiosity may be combined with other rendering techniques to achieve striking and innovative results. In the award-winning film *Bunny*, radiosity was used to render the set and the props while the two main protagonists were ray-traced and later composited into the environment as if they were live action models (Figs. 6.7.4 and 6.12.4). The direct lighting on the characters does not include any radiosity, because rendering fur with radiosity techniques would have been impractical. An interesting side effect of using radiosity was achieved by limiting the number of times that light was allowed to bounce within the environment. In this case, the one or two cycles that light bounced were not enough to resolve all the detail in the environment because there was not enough radiosity sampling. Interestingly enough, the resulting image artifact yields a look that is similar to the emulsion grain on photographic film. The creators of the film took advantage of this stochastic sampling artifact and made it serve an aesthetic purpose. The radiosity grain varies from shot to shot, and it varies based on many factors.

Photon map rendering is another version of global illumination. **Photon maps** are collections of small energy packets that are emitted into the scene to represent the way light travels through space. The values of photons as they are reflected, absorbed, or transmitted through surfaces and volumes are used to compute global illumination (Fig. 8.3.2). Caustics are a striking lighting effect rendered with this technique. **Caustics** are created when specular light is focused or dispersed by reflection or refraction, and are typically seen when light travels through crystal or water (Figs. 8.1.6 and 8.1.7).

6.7.4 The set and props in Chris Wedge's *Bunny* were rendered with radiosity techniques, while Bunny and the moth were ray-traced. In lighting this shot a bounce card was placed above the sink (and out of the frame) to emphasize the light from the lightbulb and to help focus it on the sink. Notice the film-like grain that results from limited radiosity sampling. (© 1998 Blue Sky Studios.)

6.8 Image-Based Lighting

Image-based lighting extracts global illumination information and texture data from high dynamic range photographs, and uses that information to light a three-dimensional scene. **High dynamic range (HDR)** photographs capture the full tonal detail of real-world illumination by recording a series of bracketed exposures of the same scene and merging them, usually into a single 32-bit HDR file, referred to as a light probe image. Image-based lighting can also be used to relight live performances as shown in Figures 6.8.4–6.8.6.



The image-based information includes the HDR images recorded photographically or digitally with as much quality and resolution as possible. This data gathering also includes **finite dynamic range** information about each one of the exposures and lens apertures used, samples of the intensity of light or **radiance**, information about the position of the camera and the distance to some of the key points in the scene (done with **point sampling**), and last but not least general

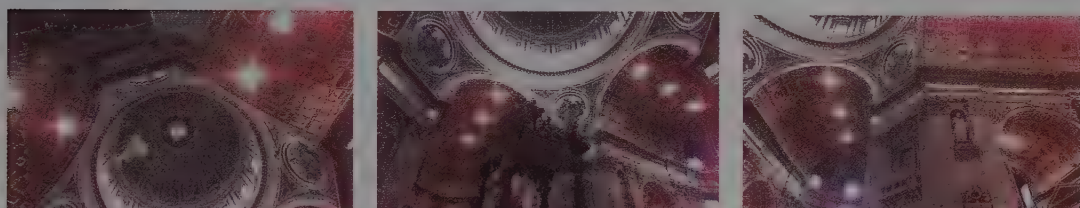
6.8.1 The rectangular slabs and the spheres in *Fiat Lux* were animated procedurally or with the dynamics simulator in Maya software. The final renderings were created on a cluster of workstations using the Radiance global illumination system to simulate the photometric interaction of the objects and the environments. The final look of the film was achieved using a combination of blur, flare, and vignetting filters applied to the high dynamic range renderings. (Courtesy of Paul Debevec, University of California at Berkeley.)

6.8.2 (Opposite page) The appearance and illumination of a real environment were recorded by taking one or two telephoto radiance images of a 2-inch mirrored ball placed on a tripod. Each of these images provided an omnidirectional illumination measurement at a particular point in space. The dynamic range of some of the resulting radiance images was stabilized, and the images processed to diminish glare. (Courtesy of Paul Debevec.)

information about the type of light that falls on the objects—for example, direct or filtered sunlight or incandescent light from bulbs. When probing and measuring light on the set, 18% gray tone matte cards—as described in Chapter 8—are commonly used as a reference for calibration purposes (Fig. 8.1.2).

A **light probe image** represents measurements of incident illumination at specific points in the real world. These measurements are omnidirectional, and are saved in the form of high dynamic range images in any number of HDR Image (**HDRI**) file formats. Light probe images can be created in a variety of ways: using a mirrored ball to capture multiple exposures of a few views (Fig. 6.8.5), stitching several high dynamic range photographs (Fig. 6.8.2), or using a scanning camera to capture panoramic views. Generally speaking the best results are obtained when using multiple high resolution panoramas that are aligned and have been compensated to eliminate radial **vignetting**, the effect in an image where its edges have a diminished brightness or saturation.

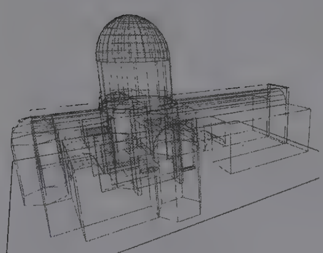
The technique of image-based lighting is closely related to photogrammetry and the image-based modeling techniques that extract geometry from photographic images. Some of the earliest experiments with image-based rendering used photographs of real environments to extract Z-depth maps of the environment and reproject pixels in three-dimensional space. The quality of these early examples of image-based renderings was very dependent on the number of views taken of the environment. A few photographs would yield gaps and errors in the rendering, but newer techniques can calculate



6.8.4 The Light Stage 6 is an image-based lighting device capable of illuminating full-body virtual characters and objects with lighting conditions from different real-world scenarios. (Courtesy of Paul Debevec, www.debevec.org.)

the missing information and fill in approximate data (Fig. 5.6.3).

The animation ***Fiat Lux*** (1999) is an example of a test-bed project created with image-based modeling and lighting techniques (Figs. 6.8.1 and 6.8.2). The geometry, appearance, and illumination of the environments were acquired through digital photography. The environments and the lighting were extracted from photographs shot on location at the Basilica of Saint Peter in Rome (Fig. 6.8.1). Different views were recorded with HDR photography in order to capture the full range of illumination; some of these seed images have a dynamic range of over 100,000:1. The high dynamic range image was made from half a dozen exposures taken with a digital camera three stops apart, and combined into a single linear response radiance panoramic image (Fig. 6.8.2). Then the props were modeled from scratch with an off-the-shelf three-dimensional modeling program that was also used to build a simple three-dimensional model of the Basilica of Saint Peter in Rome from the photographs using the Façade photogrammetric modeling system, and the image maps were projected onto the geometry (Fig. 6.8.3). Next, the position of the illumination sources recorded in the photographs was recreated in three-dimensional space, and the light probe images were used to create three-dimensional light sources of the correct intensity and location. The original illumination information was also used to remove the highlights, reflections, and ground shadows in the original photographs, allowing the synthetic objects to cast shadows and appear in reflections throughout the final rendering.

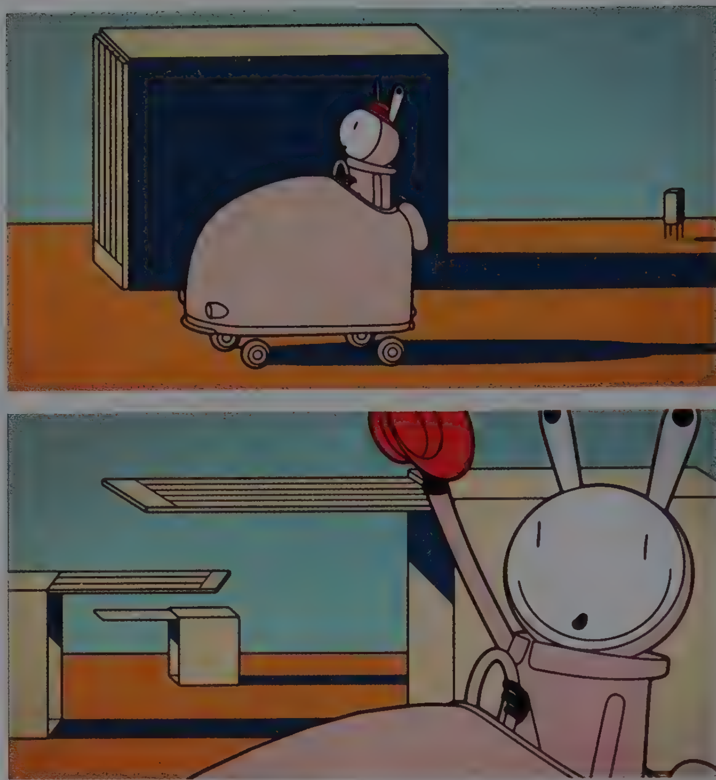


6.8.3 Geometry of the Basilica of Saint Peter in Rome used to render *Fiat Lux* (see Fig. 6.8.1). (Courtesy of Paul Debevec.)

6.8.5 (Top) Three different light probe spherical images are used to relight live performances that are composited back into the image-based lighting environment. (Courtesy of Paul Debevec.)

6.8.6 An actress performance is lit in the Light Stage 5 with 152 lighting directions and recorded by a high speed camera every 24th of a second. The recorded images can be used to relight her in new lighting environments, or to change the reflectance characteristics. The gray board is used for matting the foreground image. (Image courtesy of Paul Debevec, www.debevec.org.)

6.9.1 A boy rendered a non-photorealistic shader that applies a flat color to entire areas (top). The landscapes are rendered by grouping the geometry in different Z-depths. Notice the variable width line used to outline the shapes. (Based on the artwork of Marc Tetro. Directed by Phillip Stamp. © Tube Nunavut Inc. © Marc Tetro.)



6.9 Non-Photorealistic Rendering

Non-photorealistic rendering techniques became increasingly popular during the late 1990s to create three-dimensional computer animation and still images that look as if they were created with traditional techniques. **Non-photorealistic rendering**, or **NPR**, techniques calculate the amount of light that reaches three-dimensional surfaces but the shading is done through simulations of how traditional materials like pencils, ink, and paint pigments are distributed on a surface. There are many approaches to non-photorealistic rendering and the artistic experimentation continues to grow in sophistication and variety of styles (Figs. 6.9.1–6.9.5). Many NPR techniques are variations of three-dimensional rendering techniques and some are based on two-dimensional image processing techniques. The latter are essentially post-processing filters that can be applied to a three-dimensional model after it is rendered with realistic techniques (Fig. 14.2.6). There are also NPR shaders that combine three-dimensional and two-dimensional techniques.

Non-photorealistic rendering techniques are sometimes called **toon shaders**, or cel shaders, because they are reminiscent of the comic book drawings with a black outline and somewhat flat colors. Some hybrid renderers are able to combine realistic and NPR tech-

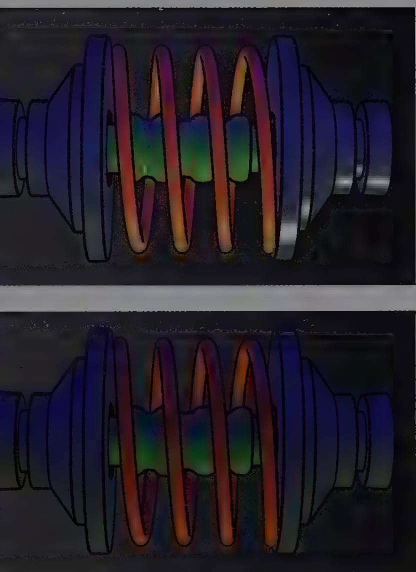
6.9.3 Black and white graphic shader applied to three-dimensional geometry to recreate the work of illustrator Charles Burns for *Fear(s) of the Dark*. See Figures 9.2.1 and 10.3.7 for additional details. (© Prima Linea Productions, www.FearsOfTheDark-themovie.com.)

niques in a single pass or multiple passes. Figure 6.9.1 shows a test comparison of simple NPR flat color rendering, and three final frames with a variable width outline and narrow color palette. Figure 6.9.2 shows a similar looking result achieved with a different approach.

Sometimes NPR shaders have to match existing styles established during the visual development stage. Figure 6.9.3 shows a high contrast look developed after much iteration to perfectly replicate the graphic style required, and a breakdown of this shader can be seen in Figure 9.2.1. The pioneering animated short *Fishing* was rendered in a watercolor simulation process that began by creating a two-dimensional matte from the three-dimensional target object (Fig. 1.4.9). Within the matte the different behaviors of fluid watercolor pigments and binders are simulated on surfaces that have different absorbency and reflectance qualities. With this process geometry can be rendered with some of the effects typical of watercolor painting including superimposed glazes in complementary colors (as in Fig. 1.4.9), dry brush effects, the edge darkening typical of wet-on-dry brushstrokes, the backruns that happen when a puddle of water spreads back into a damp region of paint, granulation and separation of pigments, and the flow patterns typical of wet-in-wet painting. The final effect was varied over time to reflect the changing time in day as well as the mood of the main character.

6.9.2 (Previous page, top) *Bunnies* was rendered with a single light source using the default toon shader in Maya software. The surfaces were rendered each with a mask to allow for the final surface colors to be set and adjusted in compositing. See page 201 for related images. (Image courtesy of Studio Soi.)

6.9.4 A toon shader gives *Paths of Hate* the cross-hatching strokes and color gradations typical of Silver Age comic books. (Copyright by Platige Image. All rights reserved.)



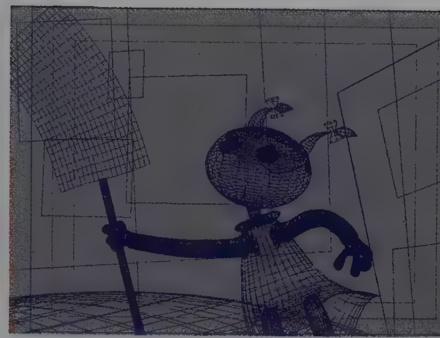
6.9.5 Non-photorealistic rendering showing the results obtained with a new shading model with highlights, a cool-to-warm hue shift and edge lines (top), and an approximation of this technique using conventional Phong shading, two colored lights, and edge lines. (© 1998 Amy Gooch.)

Figure 6.9.5 shows two variations of a nonrealistic rendering technique that changes the color hue along with the light shading information and also adds an outline edge similar to the drawing outlines used in hand-drawn traditional animation. The variations show shading and highlighting variations achieved with different versions of shading models. The rendering in Figure 6.9.6 has the look of gouache paint, and uses transparency maps to map irregular shapes like the broom onto rectangular geometry.

6.10 Hardware Rendering

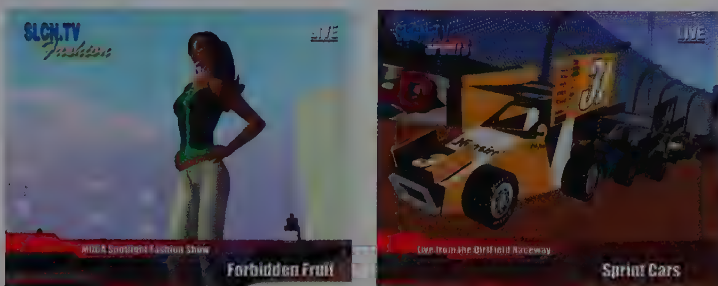
Complex rendering operations have been, and still are for the most part, traditionally calculated by the software in batch mode. This was due to limitations in early graphics cards that were initially capable of instant but limited **hardware previews** (Fig. 12.5.7). But the hardware rendering of today is no longer limited to simple shading techniques. Today's high-end graphics cards are capable of creating increasingly sophisticated renderings in real time, to the point that a few creators are producing final renders in hardware (Fig. 6.10.1).

Today's powerful graphics cards—such as those manufactured, for example, by Nvidia and ATI Technologies—add significant rendering power to computers and game platforms (Fig. 6.10.5). Take, for example, the impressive real-time rendering capabilities of some of today's popular 7th Generation game platforms. For example, the Sony PlayStation 3 renders 275 million **triangles per second** with a single-3.2 GHz Cell processor and an RSX Nvidia custom graphics card. The Microsoft Xbox 360 renders 500 million triangles with three 3.2 GHz Power PC processors and an ATI custom graphics card (Figs. 1.2.10, 10.3.8, 11.3.4, and 12.7.4–12.7.6). These are by all means impressive specification, especially when compared to those



6.9.6 Zoe the girl is rendered in a flat style that resembles *gouache* painting. (© 2002 Sparkling*, CGI by Sparx*.)

6.10.1 This scene from *Kuhfo* was animated in Maya, and rendered with Nvidia's real-time renderer Gelato. (© 2006 Filmakademie Baden-Württemberg, Holger Wenzl, Hannes Appell, Sebastian Schmidt.)



6.10.2 Fashion and sports news segments from the *Second Life Channel News TV* (© SLCN TV. Images courtesy of SLCN TV.)

6.10.3 Two shots from a *Virtual Macbeth* generated in *Second Life* with machinima tools. (© 2007 Mechtild Schmidt. Courtesy of Machinima Filmmaking in Second Life.)

6.10.4 Three real-time renderings of reddish hair geometry illustrate the realistic dual scattering approximation technique, at 1600 x 1200 pixels of resolution on a GeForce 8800 GTX graphics card. The large size rendering presents the real hair color by calculating the multiple scattering of light between hair strands: single, global, and local multiple scattering. The upper-left rendering uses only single scattering, and renders the incorrect hair color with slightly colored highlights. The lower-left rendering uses both single scattering and a global multiple scattering component. The multiple scattering of light within the volume of hair strands is what gives hair its natural color. (Images courtesy of Cem Yuksel.)

of the previous models just a few years back: the Sony PlayStation 2 rendered 66 million triangles per second; Microsoft's Xbox rendered 125 million triangles, and Nintendo's GameCube rendered 12 million triangles per second (Figs. 6.10.8 and 6.10.10). Online, machinima, and even mobile phone applications also take advantage of the graphics cards' increased power (Figs. 4.7.1–4.7.9, and 6.10.2–6.10.10).

Today's graphics cards are capable of such rendering complexity because many advanced rendering algorithms have been built into the silicon chip and also because of an increased number of microelectronic components in the card itself. The rendering power of the newer generations of graphics cards stems from a variety of technical characteristics (Fig. 6.10.6). For example, a dedicated **Graphics Processing Unit** (**GPU**), programmable vertex and pixel shaders in hardware, multiple parallel rendering pipelines, multiple textures per pixel rendering pass, 128-bit per pixel or more color resolution, shadow volume rendering acceleration, billions of full-scene antialiasing samples per second, fast Z-Buffer sorting, support of high-degree complex curved surfaces, and continuous tessellation levels per polygon for dynamic levels of detail (LOD). In addition to their three-dimensional rendering features, several high-end graphics cards are also capable of processing video with real-time features like capturing video at standard and high-definition and decoding MPEG-2 files.

Some graphics cards offer programmable vertex shaders, and this makes it possible to customize the operations on vertex data.

6.10.5 A visualization of a *Lamborghini Reventon* car, rendered in real time with ray tracing and HDRI on Nvidia hardware. The geometry and textures were built with RTT DeltaGen software. (© RTT AG.)

6.10.6 This character was rendered in real time using custom shaders with an Nvidia GeForceFX graphics card. *Dawn* has 203,741 triangles, a 98-bone skeleton, and 50 morph targets for facial animation. (© 2002 Nvidia Corporation. All rights reserved.)

6.10.7 Scene geometry from *Ashen*, a 2004 game for the N-Gage player, with about 400 polygons per frame and 5,000 per map. Rendered in real time with fully textured models. The weapon in the HUD is prerendered. (© Nokia, 2008.)

6.10.8 On average, there are about 7,000 polygons per environment (not including the characters) in the *Spyro the Dragon*™ game, rendered on a Playstation game platform. (Courtesy of Universal Interactive Studios, Inc. and Insomniac Games, Inc.)

Vertex shaders are versatile and can even be used to calculate automatic skinning of a motion skeleton. **Vertex shaders** are routines in the graphics card pipeline that can process the vertex data passed on by the three-dimensional game engine. This vertex data ultimately describes the triangles in the polygonal mesh, and it includes the position of the vertex, the diffuse color, the specular color, up to four pairs of texture coordinates, and the range of fog.

In addition to their built-in hardware rendering features a few graphics cards are fully programmable. This means that custom rendering shaders may be developed and then rendered in hardware in real time (Fig. 6.10.4). The **Cg graphics language** and the CgFX file format developed by Nvidia, for example, can be used to create and save platform-independent shaders for hardware rendering. Figure 6.10.6 shows a realistic character rendered on an Nvidia GeForce FX card with shaders written in the Cg language. In general terms the shaders written in Cg are compiled, or translated, by the graphics card to the graphics API used by a particular computer system. An **API**, or Application Program Interface, is the set of low-level commands used to display images on the computer screen. **OpenGL** and **DirectX** are the two most popular APIs used today.

Many programmable hardware shaders are as versatile as some of the non-real-time software rendering. The realistic character in Figure 6.10.6 for example, includes several hardware shaders written in the Cg language. Two vertex shaders drive her motion: a branching skeletal shader where the body mesh is driven by the animation rig, and a blend shape shader that deforms her face based on control parameters. A skin shader uses a complex combination of color maps, specular maps, and blood characteristic maps to produce human-like realistic skin. This shader calculates the oiliness of the skin surface and the amount of blood that runs just beneath the surface. Lighting subtleties are accomplished with a series of maps for diffuse specular and highlight skin lighting. An anisotropic hair shader calculates different values for hair strands depending on their orientation and external stimuli. A translucent wing shader modifies the color reflected off the wings as well as the amount of light passing through the wings based on viewing and light angles.

6.11 File Formats for Rendered Images

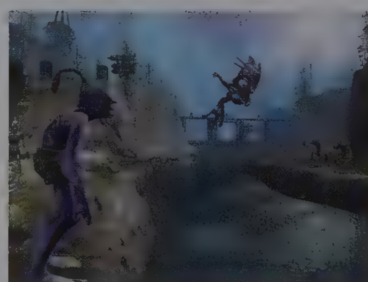
All rendering programs can save and retrieve rendered images—also called **picture files**—in their own native file format. Some rendering programs also have the capability of saving rendered image files in one or more standard file formats. This capability enhances the software's ability to share image files with other application programs or across computer platforms. For that reason we shall focus here briefly only on the file formats that can be used to exchange image files with other programs or with other computers (Fig. 6.11.1). Consult your software's manual for information about its own native rendering file format, and see Chapter 15 for additional information on file formats.

Image or picture files are not to be confused with model files. Image files contain mostly two-dimensional information and are manipulated in three-dimensional space usually as image or environment maps. In addition to their own native file formats, the majority of rendering programs offer most of the popular image file formats including: TIFF, TGA, JPEG, QuickTime, EPS, HDR, OpenEXR, and Cineon. Each of these file formats has been designed with a specific goal in mind and, therefore, each is suited to a particular task (Fig. 15.3.1). It is also possible to translate files from one format to another. Sometimes this can be accomplished within rendering programs that have internal picture file conversion capabilities. But especially when large amounts of picture file conversion are needed, it is common to use a specialized file conversion program like Adobe Photoshop or one of several standalone utility conversion programs.

Generally speaking, in an everyday production environment and when translators for exotic file formats are not available or do not work, the following standard formats would be good choices. For example, the TIFF format for best halftone detail quality, the EPS format for high-quality line drawings of wireframe renderings, the QuickTime format for compressed and portable animated sequences, and the Cineon format to store wide dynamic ranges. In theory—and most of the time in practice too—these file formats are portable both across platforms and operating systems (Fig. 6.11.2).

The **TIFF** file format (Tagged Image File Format) is popular with prepress and publishing software and useful when the rendered image will be reproduced in a publication. The TIFF format preserves detailed grayscale information that can be useful for generating the high-quality halftones (grids of dots of varying size) commonly required in publications. TIFF files tend to be large in size so many applications usually provide options or utilities for compressing and uncompressing them. The **TGA** file format is very popular with video-oriented software and is efficient and quite convenient for transferring files into the video environment. TGA is short for TARGA, the name of a family of graphics boards developed in the early 1980s that pioneered video input and output with microcomputers. The **JPEG** file format uses a very efficient image compression technique, which can affect the resolution and quality of the image (Fig. 15.3.3). Because of their compact sizes JPEG files are commonly used to transfer low-resolution images through the Internet. The **QuickTime** file format is useful for saving still images, animated sequences, and sound in a variety of levels of image compression and quality (Fig. 15.3.4). QuickTime also supports different rates of frames per second, video streaming, and sound MIDI compatibility. The **EPS** file format, or Encapsulated PostScript, is popular in prepress applications, and can be quite useful and effective when high-quality line wireframe drawings are needed. EPS files usually require significant amounts of memory for storage and transfer. The **HDR** file format is a 32-bit format used to store High Dynamic Range imagery that is essential to image-based rendering. The **OpenEXR**

6.10.9 Smoke effects, motion blur, lens glows, and flares were applied in this scene from the *Command and Conquer 2* computer game. Notice the bump and displacement maps, and the hard-edge shadow on the foreground terrain. (© Westwood Studios. All rights reserved.)



6.10.10 This screen from the game *Oddworld: Munch's Oddysee*, is rendered in real time on an Xbox game platform. (© 2003 Oddworld Inhabitants, Inc. All rights reserved. Some ™ and/or ® designate trademarks or registered trademarks of Oddworld Inhabitants, Inc. in the United States and/or other countries.)

6.11.1 Image files, such as this still from *Waltz with Bashir*, are translated from one format to another during the course of production. Using lossless file formats preserves image resolution. Notice the fully rendered three-dimensional environments behind the two-dimensional characters. (Courtesy of Bridgit Folman Film Gang. Illustrator: David Polonsky.)

File Formats for Still Images	
.bmp	Bitmap
.cin	Cineon
.eps	Encapsulated PostScript
.gif	GIF
.jpg, .jpeg	JPEG
.pic, .pict	Picture
.qt	QuickTime
.tga	TARGA
.tiff	TIFF

6.11.2 Some of the most popular file formats (and their name extensions) for still images are listed here. Read Chapter 15 for a full description of these and other file formats.

6.12.1 (Opposite page) This dialog box illustrates the importance of checking the global preferences of the rendering program before starting the rendering of a model. (© Alias|Wavefront, a division of Silicon Graphics Limited.)

file format provides a high dynamic range based on a 16-bit floating point log color space. The **Cineon** uncompressed file format is commonly used to capture the subtlety and dynamic range of images originally recorded on motion picture film, and it typically allocates 10 bits of color depth for each RGB channel. (Read Chapter 15 to learn about additional image file formats.)

6.12 Getting Ready

Rendering can be a complex process because of the large number of factors that play a role in it. That is not to say that rendering is necessarily complicated. Most of the individual steps in the rendering process are actually simple. But when the steps are all taken into account and added up, the result requires a thorough understanding of the process and some basic management skills to keep track of all the subtleties and implications buried in our choices of tools and techniques. These simple recommendations are meant to make the rendering process as successful and effective as possible.

Take Notes and Don't Throw Them Away

Even the simplest three-dimensional scene contains a large number of variables that define how the rendered objects will look when the rendering is calculated by the computer. It is always useful to write down all or some of these numbers in a notebook or a production log. It is also important not to misplace this notebook and to have it handy during the rendering process. Writing the rendering information on a small piece of paper is not good practice, since this can easily be lost. Sometimes while a scene is being rendered by the computer we want to review the information we fed earlier into the program. The rendering log comes in handy when the computer system cannot be interrupted for us to browse through the information we need while the program renders, or in cases when our access to the computer is limited; for example, the system is being backed up and cannot be accessed, we might be traveling and the batteries of our portable computer are out of power, we are at our client's location and their Internet connection to our computer network is interrupted, or we are at the beach for the weekend, thinking about our project with just a bathing suit and a towel at hand.

Rendering Is Related to Modeling

As explained throughout Section II of this book, many of the decisions we make during the modeling process have a direct impact on the performance of the rendering software. In general, and under the same rendering conditions, models that were built properly render quicker than models that were built clumsily. When we build the models ourselves we can retrace our steps and diagnose the cause of rendering problems that might originate from inadequate modeling

decisions. But it's a different story when someone else modeled the geometry—we need to find out from them whether the rendering requirements were taken into account during the modeling process.

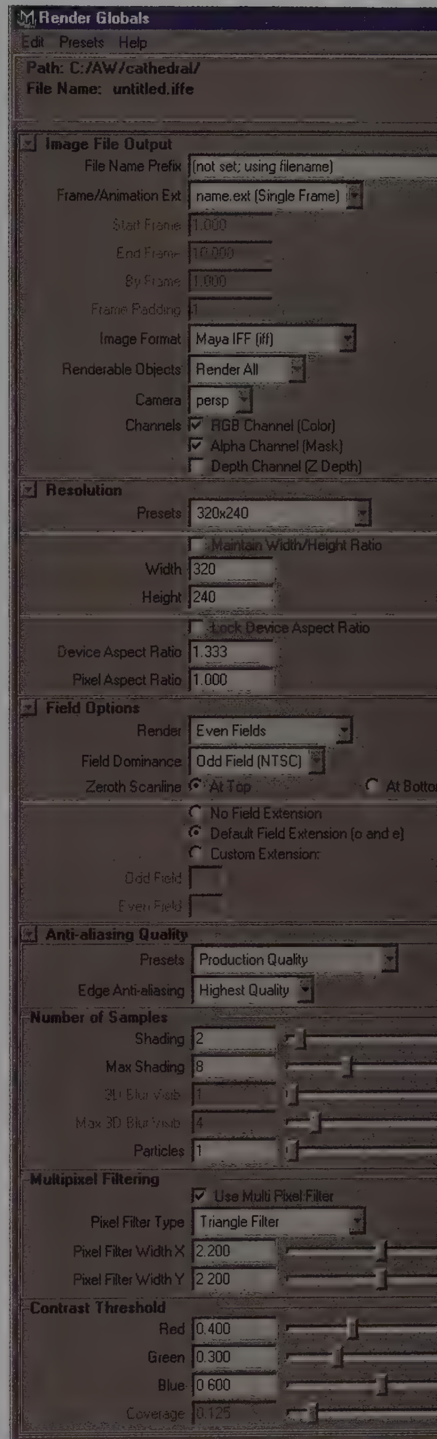
Modeling issues that can be the source of rendering headaches include intersecting, concave or open polygons, messy UV maps, poorly defined topology, small cracks or holes between surfaces that are supposed to be aligned, geometry with too many control vertices or polygons, models that were exported between different software and lost some original modeling attributes during the translation process, and objects containing other objects that are not supposed to be there. It is not uncommon to have to go back to the modeling stage, fix the modeling problems, and then return to the rendering stage with a proper model file.

Rendering Is Related to Animation

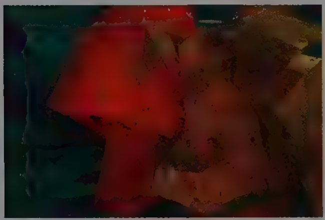
There is a big difference between rendering just one view of a three-dimensional scene and rendering thousands of frames of that scene as part of an animated sequence. When rendering a single view of a scene we can easily overindulge in rendering sophistication—for example, complex lighting arrangements, multiple layered shaders, and a deep level of ray tracing. If the render does not look good, we can try it again, ten or twenty times until we get it right. But when we work on hundreds of frames we have to choose rendering settings that look good and that can be completed by our system within our deadline. When choosing the rendering specifications for an animation, be conservative and consider the capabilities and rendering speed of your equipment. In estimating the total rendering time required, test how long it takes to render one frame, multiply it by the number of frames that need to be rendered and add another 30% time for unexpected challenges. Choose a frame that is representative of the rendering complexity in the animated sequence. It is common for the rendering specifications to vary throughout a sequence. For example, lights could be added or complex models could enter halfway through the scene.

Rendering Is Related to Output

Modifying a finished rendering is sometimes easy, when just a single pass or layer needs to be changed to fine-tune a small detail. But having to deliver the job at a different output resolution, for example, could require a total re-render. Before completing a rendering—especially one that is expected to require a lot of computer processing—it is wise to consider all the output and pipeline requirements and choose the most appropriate one. Output issues, as explained in Chapter 15, almost always have to do with issues of color, spatial considerations, and format resolution of the rendered image or sequence. Ask yourself how many images need to be rendered. One or one thousand? Is the project being rendered for theatrical projec-



REFLEC: 1, REFRAC: 0, SHAD: 0



REFLEC: 5, REFRAC: 3, SHAD: 1

6.12.2 These two versions of the same ray-traced image are based on different parameters of depth for reflection, refraction, and shadows. The image with low parameters appears more faceted and with higher contrast in the texture maps, while the second image appears more detailed and with richer textures.

6.12.3 *Viewfinder* is a method that allows users to find their photographs as overlays in a three-dimensional virtual world such as Google Earth. (USC Interactive Media Division and USC Institute for Creative Technologies. Image courtesy of Michael Naimark.)

tion or primarily for being viewed on a television screen? Are some of the images likely to be repurposed for viewing on the small screens of mobile devices?

Check the Preferences Files and File Links

If you work at home or if you have a dedicated computer at work, you might be the only person using your rendering program or your entire computer system. In that case you are probably the only one who modifies the Preferences files of the system and the rendering software. These settings control many fundamental aspects of the rendering process. As shown in Figure 6.12.1, the Rendering Preferences File may be altered in ways that impact many functions throughout the process. In addition to the Preferences files, the rendering software looks for many other references files, for example maps or shader components, that are supposed to be in particular folders. A common reason for a failed render is that the reference files are missing or stored in the wrong place.

Save Your Work Often

Save your work often, every fifteen minutes or so, and be sure to make frequent backups of your important data files. Data accidents occur when least expected, usually right around the time of a scheduled crucial delivery.

Learn the Strengths of Your Software

Every rendering software has a unique approach to select aspects of rendering. In some cases, the differences are as obvious as different names given to the same tool or different ways of presenting the information in the dialog boxes that we use for specifying values. But often the subtle differences between rendering software are significant and poorly explained in the manuals, or not even documented. In many cases it is up to you, the user, to find out some of the wonderful things that your software can do best. Explore what these features are—by using online resources such as users' groups and forums, reading wiki sites, looking for production tips and product reviews—and take advantage of them.

Consider the Limitations of Your Computer System

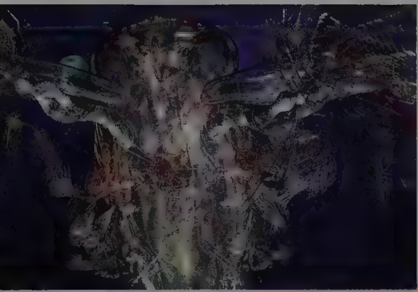
It is rare to work in an environment where hardware resources are not an issue. No matter how powerful our computers may be it is always possible to overwhelm them by submitting them to a taxing rendering. It is sensible to work within rendering specifications that are based on your system's capabilities. Unless you have an unlimited supply of funds to upgrade your computer system on a weekly basis, it makes sense to plan your work and creativity within your system's limitations.

Make Rendering Tests

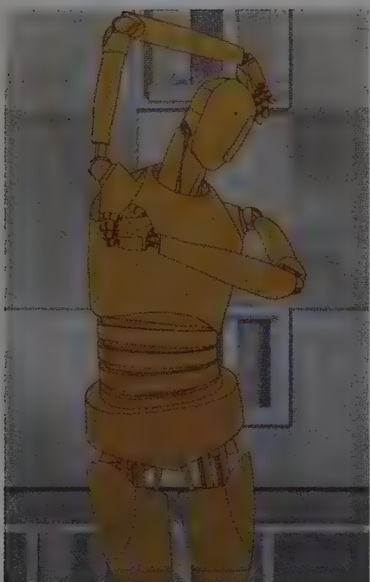
One advantage of using a computer for rendering three-dimensional models is that we can preview our work as we develop it. This ability to preview is very useful throughout the rendering process, especially when the final rendering is complex. Making rendering tests at low resolutions or with simple shading techniques is a good way to check as we go along that all of the basic rendering specifications are being applied as we want (Fig. 6.12.2). As we start applying more demanding rendering variables it is useful to test again with the full shading model, lighting rig, and spatial resolution. Finally, after we have specified the complex rendering attributes for all objects then it is time to make the last set of rendering tests to visually check everything before the final review and approval takes place and the final rendering can be produced. Making rendering tests before the final rendering will save you a lot of work later trying to fix mistakes that could have been avoided (and that might require a lot of computer processing time to fix).

Another strategy for testing rendering specifications consists of selecting only some objects to be rendered. For example, we might opt not to render some of the objects in a scene whose surface is too

6.12.4 This high angle shot of *Bunny* was rendered with a combination of ray tracing techniques for the characters and radiosity techniques for the environment and props, to produce stunning results. (© 1998 Blue Sky Studios.)



6.12.5 This still from *Invisible Ocean*, a film by François Garnier, shows how effective carefully placed light sources can be in revealing the attributes of materials, surfaces, and objects. (Executive Production: Ex Machina. © 1998 Monaco Inter Expo. Special thanks to the Oceanographical Museum of Monaco.)



6.12.6 The *Straße der Spezialisten* robot rendered with a non-photorealistic shader and ambient occlusion. See the wireframe geometry in Figure 4.5.1. (Image courtesy of Studio Soi.)

complex. Another strategy consists of turning off some secondary light sources. Usually the number of light sources in a scene is proportional to the time that it takes the computer to render the scene. Finally, a third strategy, which is supported by many rendering programs that use ray tracing rendering techniques, consists of rendering just one small area of the scene but with all objects, lights, and shading attributes turned on.

The best way to learn and really know the potential and the limitations of a rendering program is by actually using it. Additional insight can also be gained by reading the technical notes—when available—that explain how it works. Making rendering tests and comparing the results of different rendering tests while paying attention to the different variables that were used in each case is a good way to get a feel for what the numbers—or numerical values assigned to rendering variables—mean.

Optimize Your Renderings

A professional artist or technical director working in the area of three-dimensional computer rendering is defined not only by the beauty and communication power of his or her images, but also by how often he or she completes projects within the deadline. Optimizing the rendering time of an image is directly related to completing projects within deadlines, and this is done by choosing techniques that create the desired results in an efficient way (Fig. 6.12.4).

Opportunities for optimizing the rendering can be found throughout the stages of modeling and rendering stages. Use compositing techniques to assemble separate renderings into one (Fig. 9.3.7). Try using texture-mapping techniques when possible for simulating transparency, reflectivity, and roughness of a surface instead of ray tracing techniques, especially in complex scenes (Fig. 6.12.5). Experiment with non-photorealistic rendering techniques (Fig. 6.12.6). When ray tracing is a must, try to keep the ray tracing depth value down. Try keeping down the number of polygons or the geometric resolution of patches in a three-dimensional model (Fig. 6.10.10).

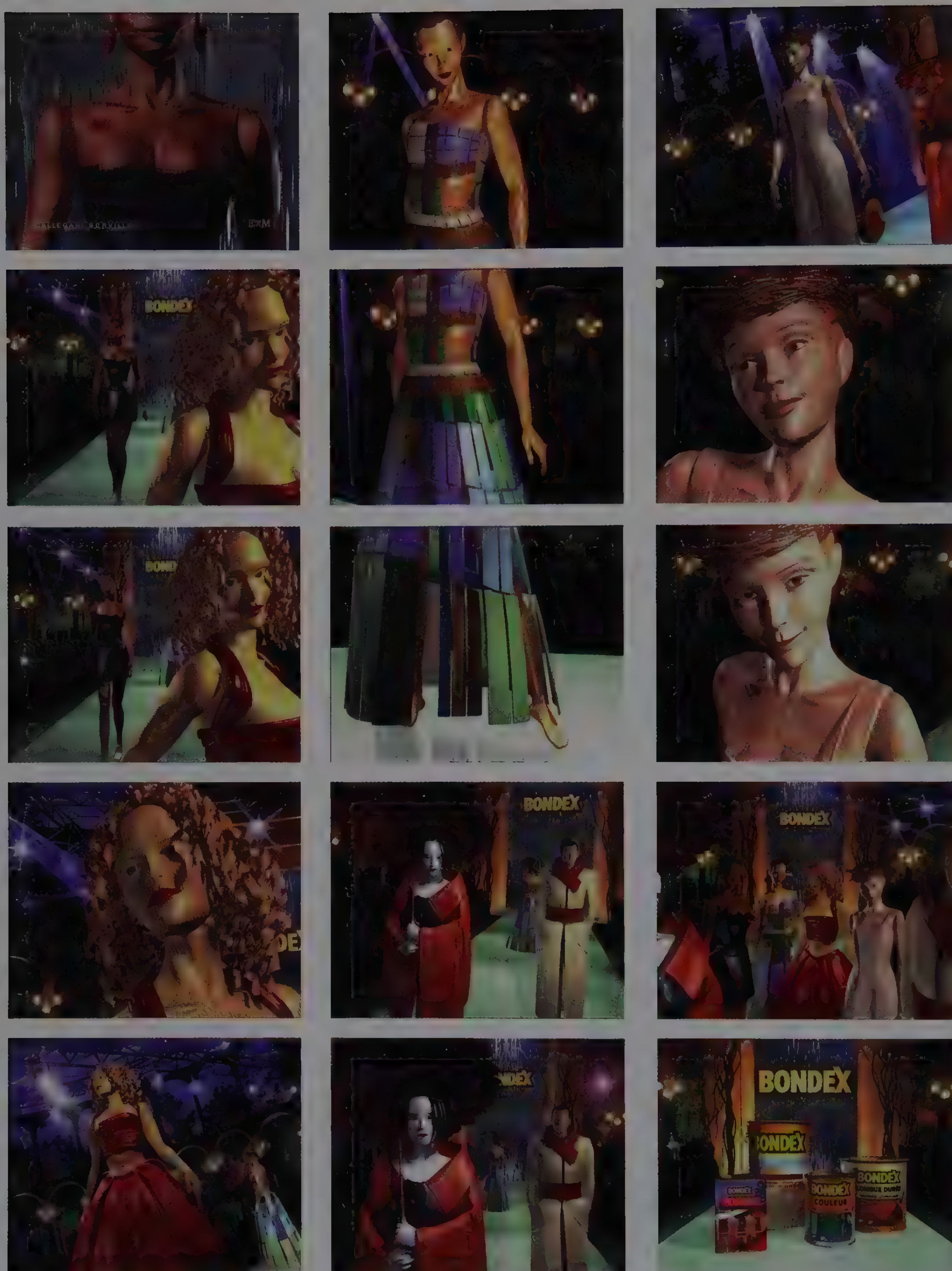
As mentioned in Chapter 8, use only the amount of light that is really necessary to create the desired mood. This principle should be kept in mind throughout the entire creative process, not only for the final rendering but also during the rendering tests created throughout. Making rendering tests before the final rendering is submitted to the computer is essential in avoiding rendering settings that might be wasteful. Try rendering critical portions of the scene before rendering the entire scene. Read Chapters 13 and 15 to learn about further optimizing possibilities by using two-dimensional techniques, and by previewing the final image in the final delivery medium and not only on the RGB monitor.

Chapter 6

Key Terms

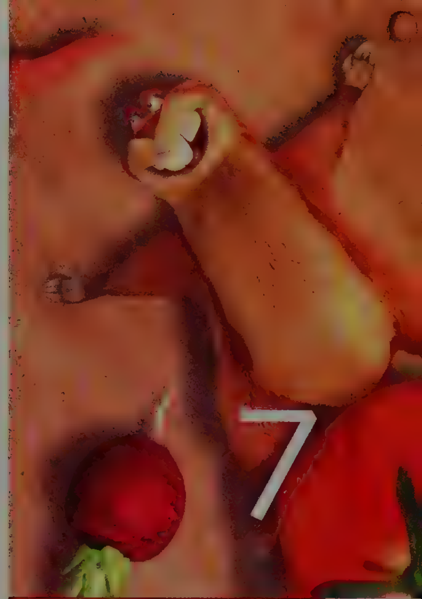
Additive color system
API, Application Program
Interface
Area subdivision
Backwards
Brightness
Caustics
Cg graphics language
Chromatic ranges
CIE color space
Cineon
CMYK, cyan, magenta,
yellow, black
Color bleeding
Color conversion
Depth sort
DirectX
EPS file format
Finite dynamic range
Global illumination
GPU
Graphics Processing Unit
Hardware preview
Hidden surfaces
HSB, HSL
HDR, HDRI, high
dynamic range image
Hue
Hybrid rendering
Image-based rendering
Interreflection of light
JPEG file format
Image space
Light-based color
Light meter
Lightness
Light probe image
Models
Non-photorealistic
rendering, NPR
Number of light sources
Number of pixels
Object space
OpenGL
Photon maps
PICS file format
PICT file format
Picture files
Pigment-based color
Point sampling
QuickTime
Radiance
Radiosity
Ray tracing, depth
Reflection rays
Refraction rays
Rendering in layers
Rendering process
RGB, red, green, blue
Saturation
Scan line
Shaders
Shadow ray
Subdivision grids
Subtractive color system
Tessellation
TGA file format
Thickness
TIFF file format
Toon shaders
Triangles per second
Vertex shaders
Vignetting
Virtual studio
Visible surfaces
Z-Buffer
Z-depth map

(The three-dimensional sets for *Bunnies* were built to match the hand-drawn layouts. The fake orthographic projection was created by building the sets to look "correct" from a single camera point of view. Looking at the sets from any other angle reveals the trick. Images courtesy of Studio Soi.)


BONDEX
BONDEX
BONDEX
BONDEX
BONDEX
BONDEX
COULEUR

CHAPTER 7

The Camera



QUICK INDEX

Summary

EMOTION AND EXPOSITION IN VISUAL STORYTELLING happen through the camera. The techniques for setting the camera within a three-dimensional scene, and for controlling and adjusting all of its parameters, are covered in this chapter. The most popular types of camera shots, moves, and lenses are also examined in this chapter.

7.1 Types of Cameras

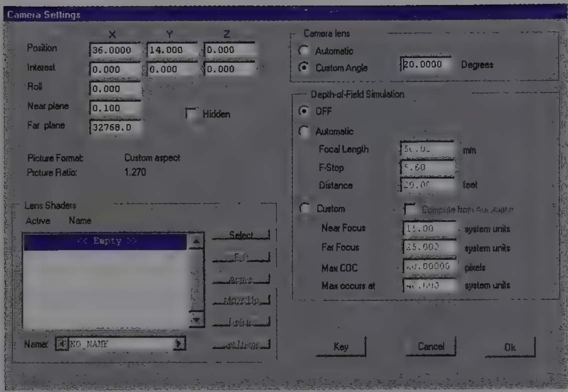
For over a century we have used cameras to select and record our reality. Throughout the years cinematographers have developed a variety of camera techniques to prioritize the elements in the frame as they relate to the flow of the storytelling. When creating a virtual three-dimensional environment we use many of those **cinematic storytelling** techniques. The composition of each shot helps the audience to understand the characters in the story and their actions in the shot. Without a **virtual camera** our computer-generated worlds and stories could not be seen or shown, let alone recorded.

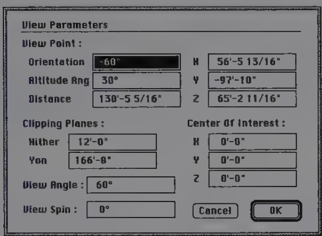
Cameras are a small but essential detail in the rendering process mostly because they define what we see in a particular shot, where we see it from, and how we see it (Fig. 7.1.1). And while many of the steps in composing the shot have to do with arranging and defining the objects in front of the camera, defining and positioning the camera itself marks the beginning of the rendering process.

In general, and for the sake of convenience, all three-dimensional rendering programs provide a **default or standard camera**. This virtual camera is usually placed not too far from and aimed at the origin (or center) of the imaginary three-dimensional world. This camera is also usually equipped with a virtual lens of medium focal length. The lens represents the scene in front of it using **perspective projection**, which projects all objects in the three-dimensional environment onto the image plane. This is done by projecting every point in space toward the camera until the projection intersects the image

7.1.1 (Opposite page) This sequence of still frames from a commercial shows how camera points of view, composition, and lighting are used to guide the viewer's eye to the desired points of interest. The placement and positioning of the camera has a hand-held feeling. (Bondex *The Fashion Parade*. Images courtesy of Ex Machina. Director: Majid Loukil. Agency: Callegari-Berville. Production: Ex Machina.)

(Top: Detail of *Gopher Broke*. Created by Blur Studio, Inc.)





7.1.2 The numerical values that define the position and characteristics of a camera can be edited through the use of dialog boxes like these. (Top dialog box from Nature FX, courtesy of Areté Entertainment, Inc.; bottom dialog box from form•Z. © 1991–1995 auto•des•sys, Inc.)

7.1.3 (Opposite page) The composition of this shot from *Polygon Family* is balanced and also dynamic. The vertical elements anchor it and the diagonals give it motion. The man walking down the street is the primary point of interest. (© POLYGON PICTURES/ IPA/NK-EXA.)

plane (Fig. 7.2.1). Other views of the default camera are commonly shown in the form of flat front, top, and side views known as **orthographic projections** (Fig. 3.6.1). The default camera can be modified or edited with mouse movements or through numerical input. Most three-dimensional rendering programs usually provide both methods. Figure 7.1.2 shows dialog boxes for editing the numerical values that define a virtual camera.

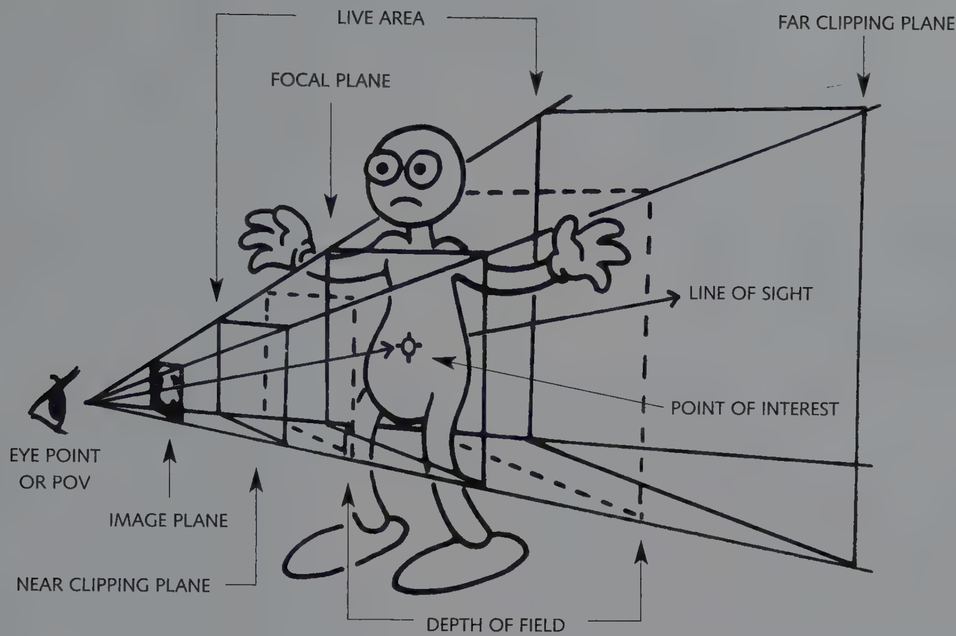
Once the default camera has been customized, it can be named just like any other object in the three-dimensional environment, and its parameters and position can be saved in a file independently from the geometry of the other objects in the world. This ability to save and retrieve the name of the camera and related information makes it possible to apply with ease any **predefined position** to the **active camera** in the scene. It is also possible to create other cameras in addition to the default camera, but when **multiple cameras** are present in three-dimensional space only one camera can be active at a time. An animated sequence or a collection of still images can be created with multiple cameras that become active one after the other as the action in the scene develops and the camera moves around or switches in sequence between different points of view.

In the wireframe display mode, cameras are usually represented with graphic icons that resemble cameras. When multiple cameras are placed in a scene, the secondary cameras placed inside of the field of vision can be seen by the main camera, unless they are made **invisible**. In many programs, all cameras are visible by default, and when the scene is rendered they appear in the image as small three-dimensional icons that usually look like little cameras.

What a virtual camera sees in three-dimensional space is defined by the type of shot, the image aspect ratio, and the type of lens. These characteristics can be set by inputting numerical values (from the keyboard or from a motion control system), or by directly manipulating the virtual camera with a variety of input peripherals that include the mouse, graphics tablet, trackball, joystick, or dial box.

7.2 The Pyramid of Vision

The pyramid of vision provides a simple way to understand some of the technical concepts involved in rendering. The **pyramid of vision**, also called the **cone of vision,** is defined as the portion of the three-dimensional environment that is seen through the camera. The pyramid of vision is defined by several parameters that are essential for controlling the position and characteristics of the camera. This numerical information includes the point of view and the point of interest, the line of sight, the near and far clipping planes,



7.2.1 The pyramid, or cone, of vision shows the visible space defined by the near and far clipping planes, and within it the depth of field area. Also shown are the image and the focus plane, the eyepoint or POV, and the line of vision representing the position of the camera and its orientation. The point of interest (POI) is represented by the target on the belly of the character.

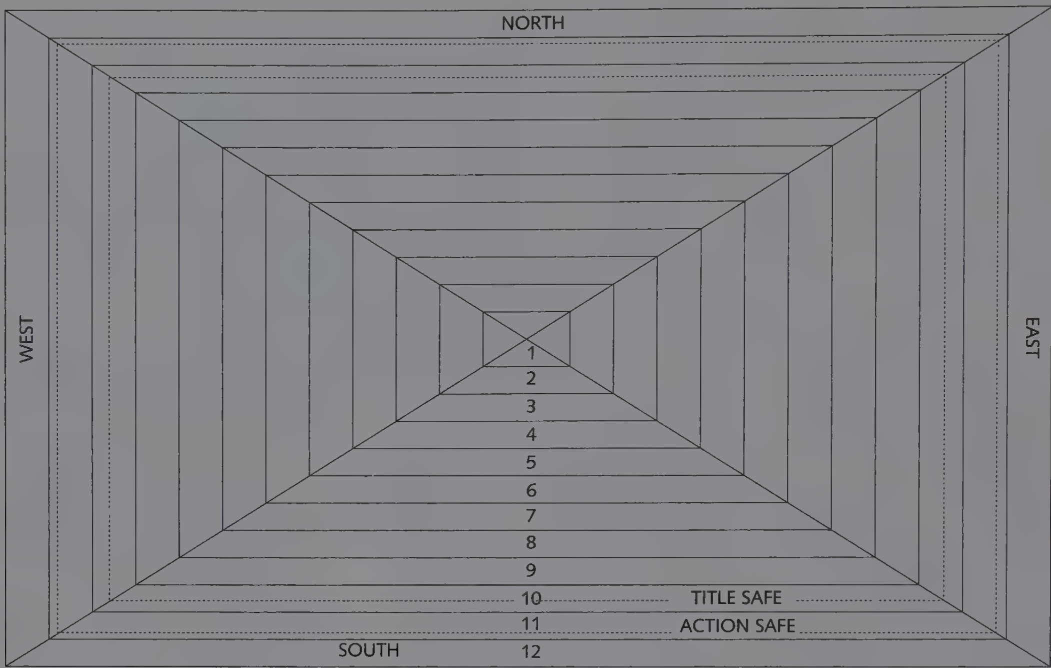
the field of vision, the viewing angle, the focal length, and the depth of field (Fig. 7.2.1). The pyramid of vision can be represented as a four-sided pyramid that grows out of the camera in the direction in which the camera is pointing. As mentioned earlier, the objects that are located inside of this pyramid can be viewed by the camera. The objects—or parts of objects—that happen to be outside of the pyramid of vision are not seen by the camera.

Points of View and Interest

The **point of view (POV)**, or viewing point, is the location in the scene where the camera is placed (Fig. 7.1.3). The **point of interest (POI)**, or center of interest, is the location in space where the camera is focused. The **line of sight** in the pyramid of vision is defined as a perpendicular line that travels away from the camera, from the point of view to the point of interest.

Clipping Planes

The clipping planes are perpendicular to the line of sight. The **far clipping plane**, also called the **yon plane**, defines the most distant area that can be seen by the camera. Think, for example, of a landscape with fog in the distance when we cannot see beyond the fog. In that case, the fog would be the far clipping plane in our field of vision (see Chapter 9 for more information on fog). The **near clipping plane**, also called the **hither plane**, represents the area closest



7.2.2 A field guide of 1:1.377 ratio (35 mm Academy format) consists of 12 concentric rectangles that help position the still elements and the action within the frame. The title safe and action safe areas for field 12 are shown with dotted lines.

to the camera that is visible to the camera. Think, for example, of your own eyelashes. Your eyes cannot see them because your eyelashes are placed before your own eyes' near clipping plane and, therefore, outside your field of vision. The **viewing angle** defines the size relation between the near and the far clipping planes. The viewing angle also defines the width spread of the pyramid of vision and, consequently, the focal length.

Field of Vision

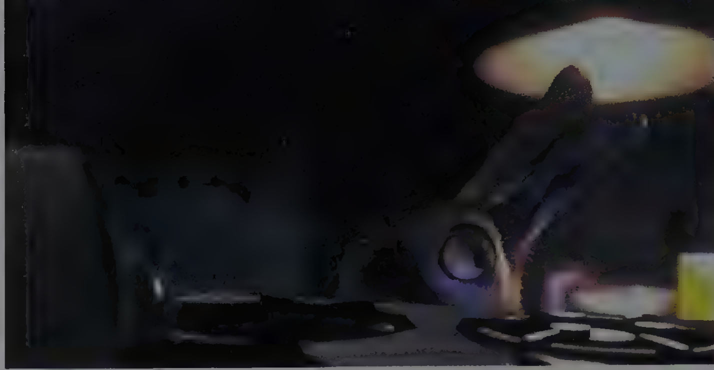
The clipping planes truncate the pyramid of vision and define the **field of vision** and the **image plane**. The objects contained inside the field of vision are projected onto the image plane to create a two-dimensional image of the three-dimensional environment. This projection process is quite similar to the way in which a real scene is projected by the optical lens used in a photographic camera onto the film that is loaded inside the camera. The relation between the width and the height of the image plane defines the **aspect ratio**, or proportion, of the image. **Media formats** such as film, video, or still photography each have their own characteristic aspect ratio. With the exception of a few square formats used in still photography—2.25 × 2.25 in., for example—computer-simulated cameras have a rectangular aspect ratio usually in a horizontal orientation, also called **landscape format**. The **portrait format**, as its name indicates, evolved as the most convenient way to frame portraits of individuals and focus on their faces and/or

7.2.3 The shallow depth of field, the tight framing, the soft lighting, and the tilt of the head all contribute to a feeling of intimacy. This is also an example of facial animation with a single model that was deformed, without blend shapes, using facial motion capture data as a guide for deformation.

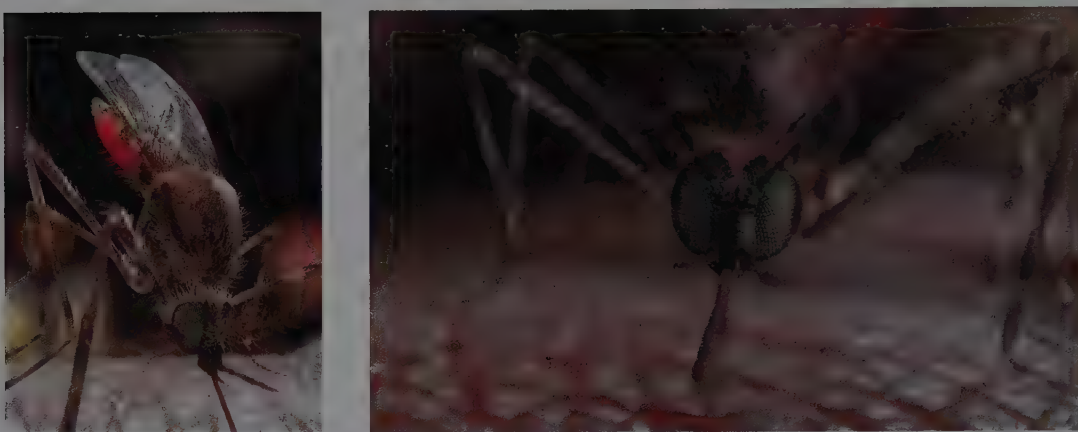
figures. Some of the most common media formats and their aspect ratios are shown in Figure 15.3.5. **Field guides** are grids of concentric rectangles used to position the still elements and the action within the frame (Fig. 7.2.2). In animation production, field guides can be used to specify zooming of the camera and to "block" the areas of the shot that can safely contain important action, titles, or credits.

Focal Length

The focal length of a camera controls the way in which three-dimensional objects are seen by the camera. The **focal length** in a virtual camera is defined by the relation between the near clipping plane and the far clipping plane. This relation defines the way in which the objects in a three-dimensional environment are projected onto the projection plane of a virtual camera—or the surface of the film in a real camera. The focal length in a photographic camera is determined by the curvature and shape of the lens, and by the distance between the lens and the image plane. Standard camera lenses have a fixed focal length, except for zoom lenses that are capable of variable focal lengths by gradually changing, in real time, the dis-



7.2.4 Notice the narrow depth of field in this low angle view of *Bunny* that emphasizes the tension and prepares the viewer for a surprise. (© 1998 Blue Sky Studios.)



7.2.5 A mosquito showing how depth of field increases the realism of a rendering. (Images courtesy of Framestore CFC and Bartle Bogle Hegarty.)

tance between the near and the far clipping planes. Virtual camera lenses can simulate almost any focal length (Figs. 7.4.2 and 7.4.3).

Depth of Field and Focus

The **focal plane** of a lens is the plane perpendicular to the camera that is resolved into a sharp image. Only one plane in three-dimensional space can be in perfect **focus** when seen through any lens, but the areas that neighbor the focal plane are in focus. The **depth of field**, also called **DOF**, is the portion of the scene in front of the camera that appears focused, and it is defined by the area between the near and the far focal planes. In renderings with a shallow depth of field, many elements appear out of focus (Figs. 4.6.3 and 7.2.3–7.2.9), while scenes with ample depths of field yield images with an overall sharpness that spans from the foreground to the background (Fig. 7.2.10). Depth of fields and focus are used to direct the viewers' attention to the relevant areas of the framed image.

7.3 Types of Camera Shots

Cameras, like other objects in three-dimensional space, can be placed in specific spatial locations in a variety of ways. The process of finding a position for a camera is called **interactive camera place-**

ment, and sometimes **navigation** because we establish the framing of a shot by navigating through three-dimensional space looking through the camera. Stationary cameras used to render still images can be placed with numerical input, interactive manipulations, or predefined positions (dynamic cameras are described in Chapter 11). Navigation is essential for framing the objects, virtual actors, and scenery in an effective way. This process can take place during or before the rendering process in order to focus on specific areas of interest and tell the story more effectively.

7.2.6 The low position and slanted angle of the camera in *They Will Come to Town* contributes to the eeriness and melancholic feeling of the foggy lighting. (© 2008 Filmakademie Baden-Württemberg, Thilo Ewers, and Holger Wenzl.)

When using the numerical input method, cameras can be positioned and repositioned by specifying two absolute XYZ locations: camera position or point of view (POV), and camera point of interest (POI). Cameras can also be positioned and repositioned using interactive mouse movements. This activates two of the basic geometric transformations: translation and rotation. All of the camera moves, even the most complex ones, can be expressed in terms of translations or rotations around one or several camera axes. But in some cases, the spherical or azimuthal coordinate system illustrated in Figure 3.4.8 is used to specify the camera's position and orientation in terms of its angles around and above the horizon and its distance from the object. (See Chapter 3 for more information on geometric transformations.)

Navigating in some programs is often accomplished by clicking buttons and dragging tools that control the camera. Other programs

7.2.7 From *Dragon Hunters*, a combination of medium and medium long shot that manages to combine full-body foreground characters with a panoramic view of the environment. (© MMVII Futurikon Films, Trixter, LuxAnimation, France3 Cinéma, RTL-Tvi, in coproduction with Mac Guff Ligne.)

7.2.8 This cleverly staged shot from *Donkey Xote* manages to combine a medium close-up shot of a principal character, and a waist shot of six additional characters arranged in two separate planes. (© 2007 Donkey Xote S.A.–LUMIQ SPA–Castelao Productions S.A.–Bren Entertainment S.A.–Don Quijote de La Mancha 2005 S.A.)

7.2.9 This medium long shot of *Birthday Boy* shows the protagonist in one of the locations where the story takes place—exteriors of the village where he lives with his mother. See Figure 4.5.5 for a close-up of the boy. (Image courtesy of AFTRS and Sejong Park.)

offer a menu of complex camera motions that can be chosen and controlled by dragging the mouse. Another technique for focusing the camera in a specific orientation consists of choosing **predefined** points of interest. These are available from pull-down menus, usually in the form of an absolute XYZ position, or an absolute angle such as X=45˚ Y=30˚ Z=60˚, or the name of an object in the scene, such as "point to the large tree closest to the house."

There are several types of **stationary camera shots**. Each one has a specific name and an inherent **narrative and psychological effect**. Most stationary camera shots can be described in terms of their point of view, point of interest, the distance to the subject, and the type of lens used. The most common camera shots are listed in Figure 7.3.1. (Animated camera moves are described in Chapter 11.)

Both the point of view and the point of interest are used to define the traditional camera shots: point of view (POV), low angle, high angle, and reverse angle shot. The distance from the camera to the subject and the type of lens used defines the area of the scene that is captured by the camera. The camera shots based on the area of the scene that is framed within the image are illustrated in Figure 7.3.2 and include extreme close-up, close-up, medium close-up, waist, knee, medium, wide, medium long, long, and extreme long shots. Complex staging of the subjects may combine several camera

positions into a single shot (Fig. 7.2.8). Most software programs use the names used in traditional cinematography to define camera shots.

7.2.10 The golden hour lighting effect typical of sunset is recreated in this shot. A tilted camera accentuates the perspective. (© NAMCO Ltd. All rights reserved.)

Point of View Shots

Point of view shots often place the camera at eye level looking straight into the action, when it is assumed that the active character is standing in front of the action. A **point of view shot** shows what the active character, narrator, or virtual camera person sees. This type of shot places the camera wherever the eyes of the active character happen to be and sets the orientation of the camera according to the direction and speed in which the active character or narrator is looking and/or moving (Fig. 7.3.2).

7.2.11 High angle shot of the main character in *Varmints*. (© Studio aka 2008.)

Low Angle and High Angle Shots

In low angle and high angle shots, the camera is pointed at the action with a certain slant. The angle is usually defined in relation to the point of interest so that a **low angle shot** places the camera below the point of interest, looking up (Fig. 7.3.2). Inversely, a **high angle shot** places the camera looking down at the point of interest, placed above it (Figs. 6.12.4 and 7.2.11). The amount of slant in

Types of Camera Shots
Extreme close-up
Close-up
Medium close-up
Waist
Medium
Knee
Wide
Long
Medium long
Extreme long

7.3.1 Using different types of camera shots gives motion pictures their rhythm, intonation, point of view, and narrative tension.

low and high angle shots is never implied and has to be defined explicitly in the form of an XYZ position or a specific angle measured in degrees. The range of low angle shots includes, for example, what a camera person would see if shooting lying down or in a kneeling position, or if shooting while standing up at street level and looking up at the action occurring on the roof of a house. High angle shots range from a camera person shooting over a crowd or perched on a ladder to a camera mounted on a helicopter hovering over a crowd.

Reverse Angle Shots

This type of shot is commonly used in dialog scenes between characters and it always happens as a response to a previous shot. A **reverse angle shot**, also called reverse shot, typically shows the action from the opposite side and in the opposite direction as the previous shot. A shot/reverse shot dynamic is ideal for building a visual narrative that goes back and forth between two characters. A common version of a reverse shot places the camera over the shoulder of one of the characters as he/she replies to the other character in the dialog scene. For example, a shot of character A framed slightly on the left side of the frame and looking to the right while talking to character B, is followed by a reverse shot of character B framed slightly on the right side of the frame and looking to the left while replying to character A (Fig. 7.3.3).

Close-Up Shots

A **close-up shot** places the camera at close range so that details in the subject can be appreciated. A close-up of a face, for example, shows nuances of expression. A close-up shot of a cut precious stone focuses on the delicate interplay of the facets and the refracted light. An **extreme close-up shot** is even closer to the subject than a close-up and presents delicate surface details such as the veins in the leaf of a plant or the wrinkles on a face (Fig. 5.2.1). A close-up shot usually fills the image with the subject in question and crops all other items in the picture (Fig. 7.2.3). A **medium close-up shot** presents subjects close to the camera while leaving some space between the subject and the edge of the frame to include a small portion of the background. A medium close-up of the face of a character is often called a **head shot** because it focuses on the face, neck, and shoulders. Head shots typically focus on facial expression and head movements (Figs. 10.3.3, 10.3.4, and 10.4.1).

(Credits for Fig. 7.3.2. Extreme close-up: © Michael Koch. Close-up: ReBoot® and © 1997 Mainframe Entertainment, Inc. All rights reserved. Medium close-up: © Jim Ludtke. Waist: © Westwood Studios. All rights reserved. Knee: © 1998 Square Co., Ltd. All rights reserved. Medium: © TeamTO–TF1–Teletoon, 2006. Wide: © 2006 A. Film, Magma, Futurikon, Ulysses. Medium long: © 2008 Kenges/Simon Bogojevic Narath. Long: © 1999 Oddworld Inhabitants, Inc. All rights reserved. Point of view: Images courtesy of Ex Machina, full credits in Fig. 10.2.2. Low angle: © Westwood Studios. High angle: © 2008 A. Film A/S.)

Medium and Wide Shots

A **waist shot** presents characters from the waist up. A waist shot focuses on the upper body language and gestures of the character, and includes a fair amount of the three-dimensional environment

EXTREME CLOSE-UP

CLOSE-UP

MEDIUM CLOSE-UP

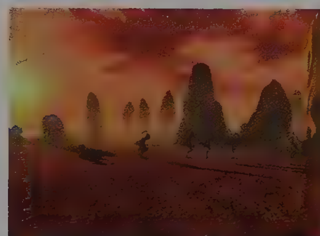
WAIST

KNEE

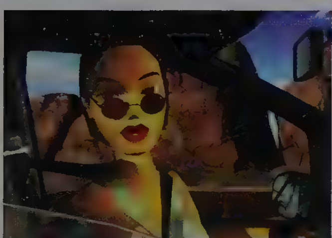
MEDIUM

WIDE

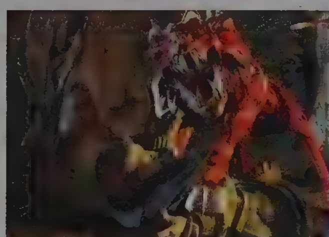
MEDIUM LONG



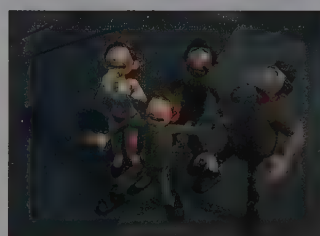
LONG



POINT OF VIEW



LOW ANGLE



HIGH ANGLE

surrounding the character. Heads and faces are never cropped in a regular waist shot (Figs. 1.3.6, 6.9.3, 10.3.6, 10.4.3, and 11.2.10). A **knee shot** crops the subjects at the knees, and is commonly used in shots of an encounter or conversation between two or three characters (Figs. 1.4.5, 1.4.8, and 7.3.2). A **medium shot** frames the characters anywhere from the hips up to full body (Fig. 12.4.2). A **wide shot** presents enough of the scene to include the full bodies of five characters (Figs. 1.3.4, 10.3.8, 11.5.4, and 12.2.2).

7.3.2 Using different types of camera shots gives moving pictures their narrative tension, rhythm, intonation, and point of view. (Credits on previous page.)

7.3.3 A reverse angle shot showing a moment in the wordless conversation between a cowboy and a mosquito. (© 2001 Angela Jedek.)

Focal Lengths of Popular Lenses	
Fisheye	7.5 mm
Extreme Wide	18 mm
Wide Angle	24 to 28 mm
Medium Wide	35 mm
Standard	50 to 55 mm
Medium Long	80 mm
Long (Telephoto)	135 to 250 mm
Extra Long (Supertelephoto)	500 mm or more

7.4.1 Typical focal lengths of popular types of lenses used in photography, cinematography, and video.

Long Shots

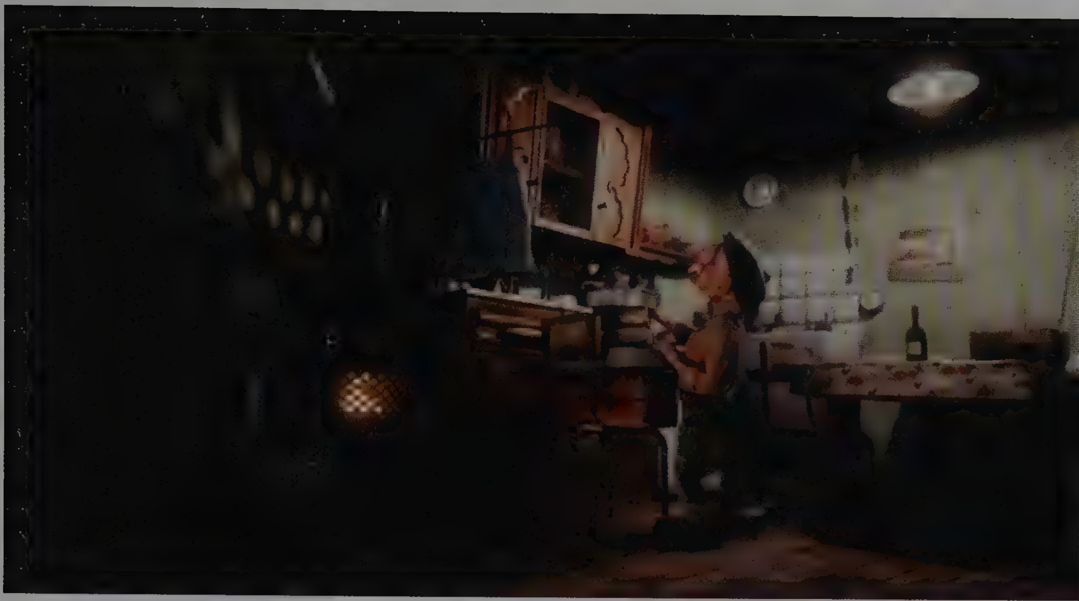
Wide and long shots are both used in animated sequences as establishing shots to introduce the place where a scene is supposed to be taking place. A **long shot** focuses on the scenery and barely permits recognition of individual characters in the environment (Fig. 1.4.1). A **medium long shot** is less open than a long shot, and it typically focuses on features like the ambient lighting, the weather, and the time of day in the scene (Fig. 7.3.4). An **extreme long shot** presents environments seen from very far away—for example, the planet Earth seen from an orbiting spacecraft.

7.4 Types of Camera Lenses

Most three-dimensional rendering software provides an infinite range of camera lenses that can be used for practical and stylistic purposes. The **stylistic use** of employing different camera lenses is to create different moods in the scene. The emotional effect of a wide angle lens, for example, is intense and can even be frightening because objects look distorted. The opposite emotion is aroused with a telephoto lens, one of tranquility and detachment, because the objects in the scene are not distorted and most of the lines in the composition are horizontal and static. The **practical use** of switching camera lenses is to modify the way in which the subjects fill the frame without having to move the camera.

Camera lenses, whether real or simulated, are perhaps the most important component in any camera system because they define the way in which the three-dimensional world is projected onto the **projection plane** of a camera. Photosensitive film, for example, is located exactly on the projection plane of photographic cameras. The projection plane of computer-simulated rendering cameras can be positioned virtually anywhere in space.

Photographers refer to camera lenses in terms of their focal length because this characteristic controls the way in which three-dimensional objects are seen by the camera (Fig. 7.4.1). But computer lenses simulated with rendering software are not limited to the standard focal lengths of photographic lenses. Focal length, as explained earlier, is defined by the distance from the point of view to the focal plane. Most photographic camera lenses have a **fixed focal length**, except for the so-called zoom lenses that contain multiple lenses and are therefore capable of a range of **variable focal lengths**. Figure 7.4.2 illustrates the effect of changing the focal length of a virtual camera by modifying the distance between the point of view to the focal plane. The scene viewed through a normal lens with a focal length of 50 mm looks similar to the way we see reality with our vision. The diagonal lines are steeper and the perspective projection is more extreme with the 28 or 15 mm lenses. The same scene viewed through a 5 mm wide angle lens looks distorted and tense. The 80 or 100 mm telephoto lenses flatten the per-



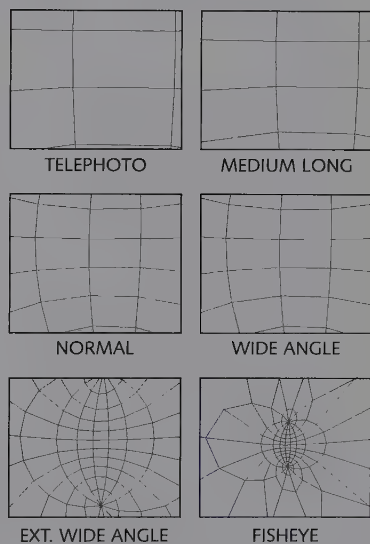
7.3.4 Wide long shot from *Même les pigeons vont au paradis.* (© BUF, Director Samuel Torneux.)

spective due to their narrow viewing angle and the similarity between the areas of the near and far clipping plane.

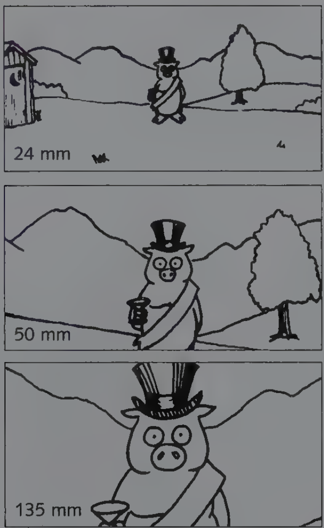
The standard nomenclature for the focal length of lenses is expressed in millimeters (mm). There is a great variety of lenses, but the staple lenses used in traditional photography and cinematography include a normal 50 (or 55) mm lens, a wide angle 28 (or 24) mm lens, and a telephoto 135 mm lens. Each of the three standard lenses has characteristics that can be used for different situations. In general, lenses with a short focal length offer a wide angle of view and increased depth of field, but objects appear distant to the camera. Inversely, lenses with a long focal length have narrow angles of view and depths of field. The relation between the focal length of lenses and their angle of view is illustrated in Figure. 7.4.3.

The area of the scene that is framed within the image can be defined by: the type of lens used, the distance between the camera and the subject, or both. In the first case, when the type of lens is varied, it is assumed that the distance between camera and subject remains constant (Fig. 7.4.3). In the second case, when the distance between camera and point of interest is changed to include more or less of the image, it is assumed that the type of lens used remains constant. Only a small portion of the scene is contained in the frame when the camera is very close to the subject. As the camera is placed farther away from the subject a larger area of the scene is contained in the frame (Fig. 7.4.4).

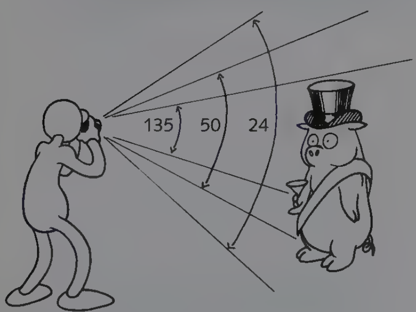
Figure 7.4.5 illustrates three different examples of what happens when both the lens and the point of view are changed. In the first example, a 24 mm wide angle lens is placed close to the subject. The wide angle view of the lens makes the shot very panoramic and



7.4.2 A camera placed inside of a sphere "sees" a different image through different lenses. The magnification of the scene viewed through the lens is proportional to the focal length.



CHANGING LENSES



7.4.3 As the focal length of a fixed lens increases its angle of view decreases.

shows a large area of the scene. In the second example, a 50 mm standard lens includes less of the background in the three-dimensional environment because its angle of view is narrower than the previous lens, but the subject occupies roughly the same amount of image area. In the third example, a 135 mm telephoto lens is placed far from the subject. Due to its long focus (or ability to concentrate on distant objects) the image retains as much of the subject as the previous lenses, but it includes little of the scenery.

Normal Lens

The **normal lens**, **50** or **55 mm** for example, offers a standard 46-degree angle of view with average depth of field. A 50 mm lens is useful for medium to wide shots because it can fill the field of vision with foreground subjects and background without flattening or distorting the perspective—as telephoto or wide angle lenses do.

Wide Angle Lens

A **wide angle lens**, **24** or **28 mm** for example, supplies a generous 83-degree angle of view, which yields outstanding depth of field. This type of lens also provides a small amount of distortion on the edges of the picture due to the forced perspective projection typical of the wide angle of vision (Fig. 7.4.6).

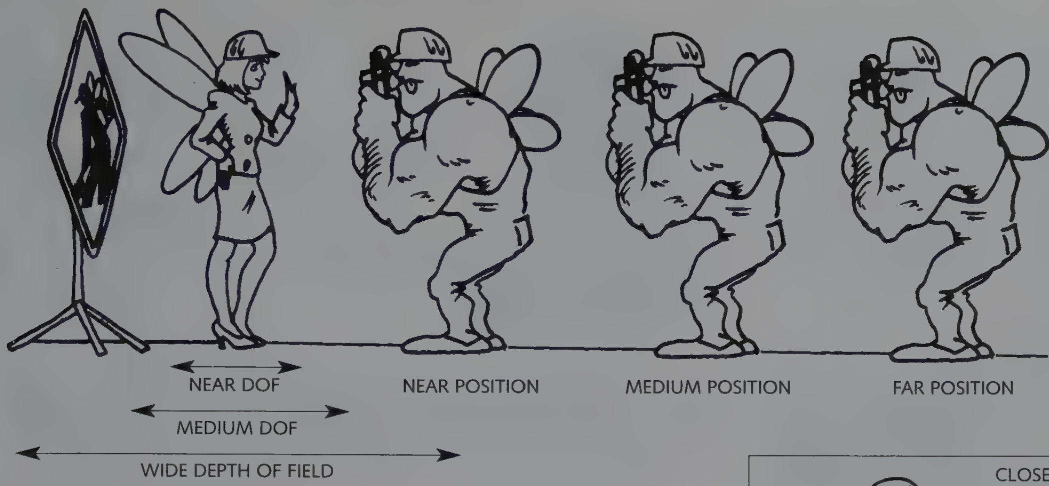
Telephoto Lens

A **telephoto lens**, **135 mm** for example, has excellent abilities for close framing. However, it flattens the perspective and has a narrow 5-degree angle of view and a small depth of field. On occasion, the wide angle and telephoto lenses can be replaced or complemented with a zoom lens that allows for a variable focal length—for example, from 35 mm to 80 mm.

7.5 Camera Animation

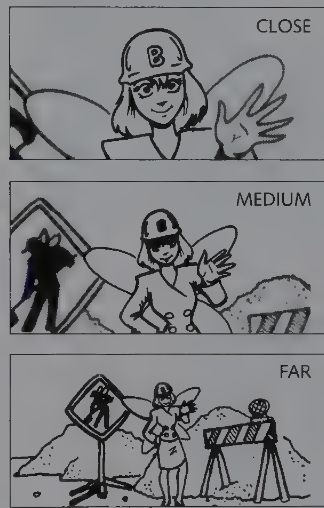
The camera has a powerful storytelling effect because it leads the eyes and minds of an audience throughout the process of visual storytelling. Animated camera moves can be based on both changes of position and orientation.

The **camera moves** that are based on a change of the **position** of the camera include a dolly, a truck, and a boom. A **dolly** is a translation of the camera along the horizontal axis (Fig. 11.3.1). A tracking or **traveling shot** occurs when a dolly moves along with the subject and follows it (Fig. 13.12.5). The motion parallax effects that can happen on dolly moves are described in Chapter 11. A **truck** move is a translation of the camera along the depth axis, and it usually goes in or out of the scene. A **boom** is a translation of the camera along its vertical axis. A **crane shot** can be implemented with



a combination of boom, truck, and sometimes dolly camera moves.

The camera moves that are based on the change of the **orientation** of the camera include a tilt, a roll, and a pan. A **tilt** is a rotation of the camera on its horizontal axis. A tilt is also called a pivot and is used to look up or look down. A **roll** is created by rotating the camera around its Z axis. Roll camera moves are common when simulating fly-throughs. A **pan** is a move created by rotating the camera around its vertical axis (Fig. 11.3.2). Panning is very effective for scanning the scene from side to side while the camera remains stationary. Sometimes, especially when simulating flying cameras, a tilt move is called a pitch—as in airplanes pitching—and a pan move is called a yaw. (A zoom is a camera move that is achieved not by moving the position or orientation of the camera but by animating its focal length.)



CHANGING CAMERA DISTANCE

7.4.4 Both the size of the image area and the depth of field increase when a camera moves away from a fixed subject.

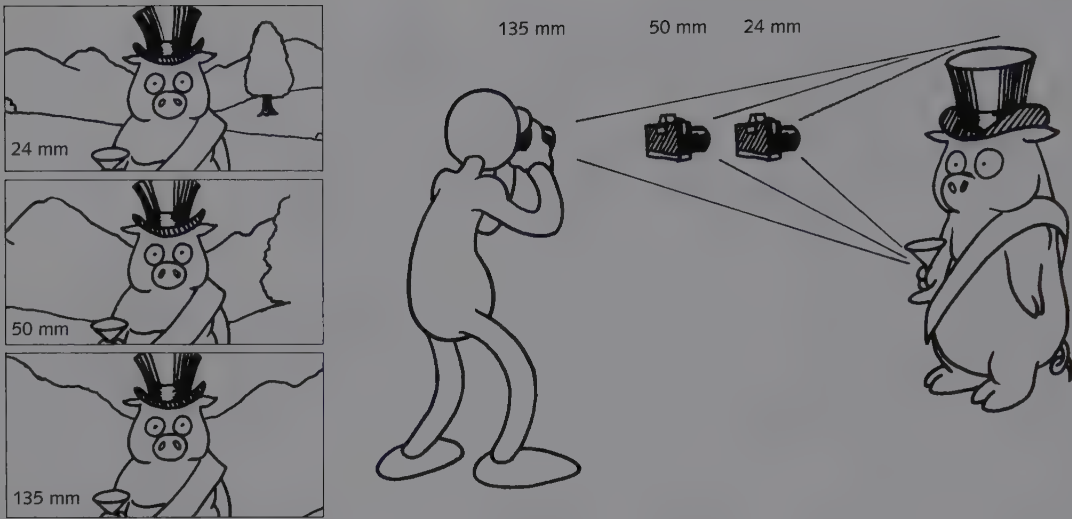
7.6 Getting Ready

Set the Aspect Ratio Early

The aspect ratio of a virtual camera determines the relation between the width and the height of the final image. It is important to set the correct aspect ratio of the image early on in the creative process because many decisions like composition and lighting are closely tied to it. Changing the aspect ratio in the middle of a production may mean that the placement of all the cameras, lights, and even the objects in the scene may have to be done all over again.

Composition Tips

When composing still images it is useful to remember that the arrangement of elements within the image frame plays a fundamental role in expressing the emotion or telling the story behind the image.



7.4.5 The pig is framed so that it occupies roughly the same amount of space in each image, but each view has been created with a different lens and distance between the camera and the pig. The subject in the foreground of the image remains within a similar scale as long as the focal length decreases or increases along with the distance between the camera and the subject. The projection of the background elements, however, is significantly different.

7.4.6 Wide angle lens distortion. (*Jak & Daxter* image courtesy of Naughty Dog, Inc. © 2001 Sony Computer Entertainment America, Inc. See page 328 for full credits.)

The following composition tips can be applied to any image, regardless of the subject. Whether a composition is simple or complex, there are some basic qualities and rules that contribute to **straightforward communication**. These include the clarity of foreground subjects, the number of image layers between the foreground and the background, the density of the background, the relation between the foreground and the background, the relation between the center of the image and the edges, and the relation between image zones and image proportions. Figure 7.1.3 shows a shot with the main character in the foreground, walking to his destination, and the secondary characters in the middle ground as a smaller but important visual element that catches our attention. The horizontal lines in the composition add some tranquility while the perspective lines add motion.

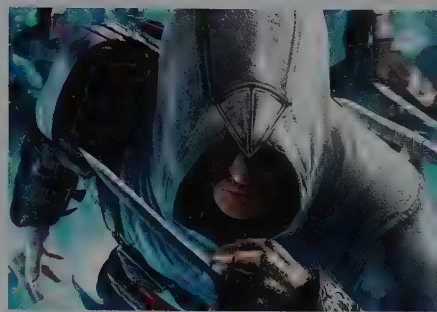
Keep the long straight lines in the composition parallel or perpendicular to the edges of the image to avoid unwanted tension and distraction. This includes, for example, the horizon or a tall tree in a landscape tilting to one side, especially when the tree is close to an edge of the image.

It is usually distracting to cut off the head of a subject in a head shot or a portion of the object in a close-up shot. However, when done skillfully, cutting off portions of the main subject can help the viewer focus on details—such as the eyes or the mouth—that may add emotion to the image.

Positioning the camera too close to an object may result in images with large unfocused areas occupied by these objects. This effect often overwhelms the rest of the image. Objects that are too close to the camera can be used to create effects of intrusion or anxiety, but they should only be used if those emotions are the right ones to present the main subject to the audience.

When image clarity is an important issue, it helps to place the

7.6.1 Two dynamic shots from *Eels*, where the framing, tilt, and roll of the camera give the scene a stylized look that helps visualize the sickness of the main character. (© Filmakademie Baden-Württemberg, Martin Rahmlow, 2006.)



7.6.2 The tilt, shallow depth of field and high contrast highlights contribute to the dynamism of this shot from the *Assassin's Creed* game. (© Ubisoft Entertainment. All rights reserved. Assassin's Creed, Ubisoft, Ubi.com and the Ubisoft logo are trademarks of Ubisoft Entertainment in the US and/or other countries.)

main subject in a shot against plain backgrounds. Backgrounds with dense textures or with a multitude of objects and colors tend to take the attention of the viewer away from the items in the foreground.

Dynamic Cameras

Unlike animated shorts or feature movies where the camera is fixed and predetermined, the cameras in virtual worlds, computer, and platform games are dynamic and allow game players to control what they see as they look around or as they navigate through the environment (Figs. 7.6.1–7.6.3, and 11.6.2). **Dynamic cameras** give players the ability to view the action from different points of view. At any given point in most games players may switch between multiple cameras usually located at the player's point of view, **first-person camera**, and other locations. The latter are usually called **third-person cameras** because they show other characters' point of view, but they might also include bird's eye views (Fig. 7.2.11), and cameras placed at fixed locations, the entrance to the dungeon for example. Most **dynamic cameras** are capable of six degrees of freedom: XYZ translations as the player moves, and XYZ rotations (tilt, roll, and pan) as the character looks around. Usually, first-person cameras have a limited rotation range—in some games for example, it is not possible to tilt down (X rotation) a first-person camera enough to see the character's feet.

7.6.3 (Next page) Screens from the game *Oddworld: Munch's Oddysee* rendered on an Xbox game platform. (© 2003 Oddworld Inhabitants, Inc. All rights reserved.)

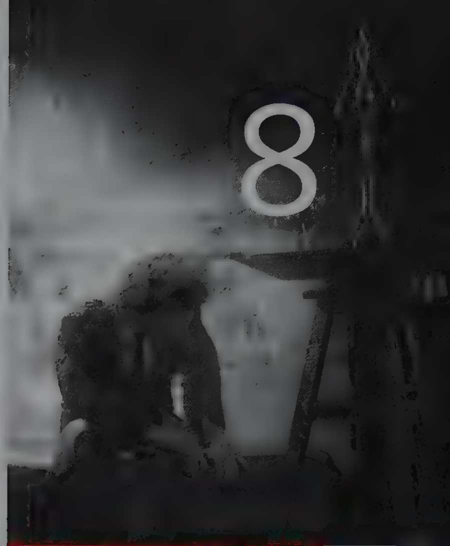
CHAPTER 7

Key Terms

Active camera
Aspect ratio
Boom
Camera lenses
Camera moves
Cinematic storytelling
Close-up shot
Cone of vision
Crane shot
Default or standard camera
Depth of field
DOF
Dolly
Dynamic cameras
Extreme close-up shot
Extreme long shot
Far clipping plane
Field guides
Field of vision
First-person camera
Fixed focal length
Focal length
Focal plane
Focus
Head shot
High angle shot
Hither plane
Image plane
Interactive camera placement
Invisible
Knee shot
Landscape format
Line of sight
Long shot
Low angle shot
Media formats
Medium close-up shot
Medium long shot
Medium shot
Multiple cameras
Narrative and psychological effect
Navigation
Near clipping plane
Normal lens, 50 or 55 mm
Orientation
Orthographic projections
Pan
Perspective projection
Point of interest (POI)
Point of view (POV)
Point of view shot
Portrait format
Position
Practical use
Predefined
Predefined position
Projection plane
Pyramid of vision
Rack focus
Reverse angle shot
Roll
Stationary camera shot
Straightforward communication
Stylistic use
Telephoto lens, 135 mm
Third-person cameras
Tilt
Traveling shot
Truck
Variable focal lengths
Viewing angle
Virtual camera
Waist shot
Wide angle lens, 24 or 28 mm
Wide shot
Yon plane

CHAPTER 8

Lighting



Summary

THIS CHAPTER DESCRIBES THE MAIN ELEMENTS of lighting a shot, presents a variety of simple and complex lighting strategies, and covers some of the basic techniques for controlling and adjusting the lights that illuminate the environment. Lighting is an important component of the rendering process not only because it may contribute significantly to the overall processing time necessary to render the scene, but mostly because it reveals the three-dimensional world and sets the mood of the scene.

QUICK INDEX

8.1 Lighting Strategies and Mood

There are as many philosophies of lighting as there are disciplines that require lighting. This would include the performing arts—dramatic theater, musical theater, dance, opera—which usually take place indoors, and cinematography, which may take place both indoors and outdoors. Lighting designers in each of these disciplines favor particular approaches to lighting, which in turn are based on the discipline's lighting needs. Of course, within each discipline there are many different points of view. We can learn a lot of interesting lighting techniques and creative points of view by examining specific movies, plays, operas, and musicals. Next time you have the chance to attend a performing arts event, pay attention to the lighting arrangements and try to figure out how they affect the mood of the moment.

Much of the mood in any computer-generated scene is also established with the choice of lights and their arrangement (Fig. 8.1.3). Lighting can be bright and fresh, soft and intimate, multicolored and festive, or tinted and moody. Light on a scene can be even and peaceful or uneven and disturbing. The shadows created with light can be harsh and sharp or soft and slow. Many of the figures in this chapter show different moods that have been achieved mostly with the effective use of lighting. Many techniques for describing, measuring, and arranging light in the scene have been developed

(Top: Detail of *Eternal Gaze*. © 2008 Sam Chen and Aloha Animation.)

Lens Aperture f/stops	
f/1	f/11
f/1.4	f/16
f/2	f/22
f/2.8	f/32
f/4	f/45
f/5.6	f/64
f/8	f/90

8.1.1 This scale lists the f/stops used to measure the aperture of the lens, and to determine how much light will pass through it. With each additional f/stop, the amount of light passing through the lens doubles. f/1 represents the lens wide open and f/90 is only a small aperture.

The Zone System	
Zone 0	3.5%
Zone I	4.5%
Zone II	6%
Zone III	9%
Zone IV	12.5%
Zone V	**17.5%**
Zone VI	25%
Zone VII	35%
Zone VIII	50%
Zone IX	70%
Zone X	100%

8.1.2 The zone system is used to catalog the gray levels on a scale that goes from nonreflective black to absolute white. Zone V represents the average reflectance of objects.

over the years in traditional still photography and cinematography. In addition to its essential artistic and storytelling purpose, lighting is also paramount in the correct exposure of the scene whether it is recorded on film, video, or digitally. As of yet, only a few of these basic traditional techniques have been translated into the mainstream of computer-generated lighting. While the general principles of lighting for a live action film, for example, are almost identical as those of a computer animation feature, the processes and subtleties of the craft are sometimes far apart. But this gap is slowly closing as we continue to develop both the physical and software tools.

Visualizing Light

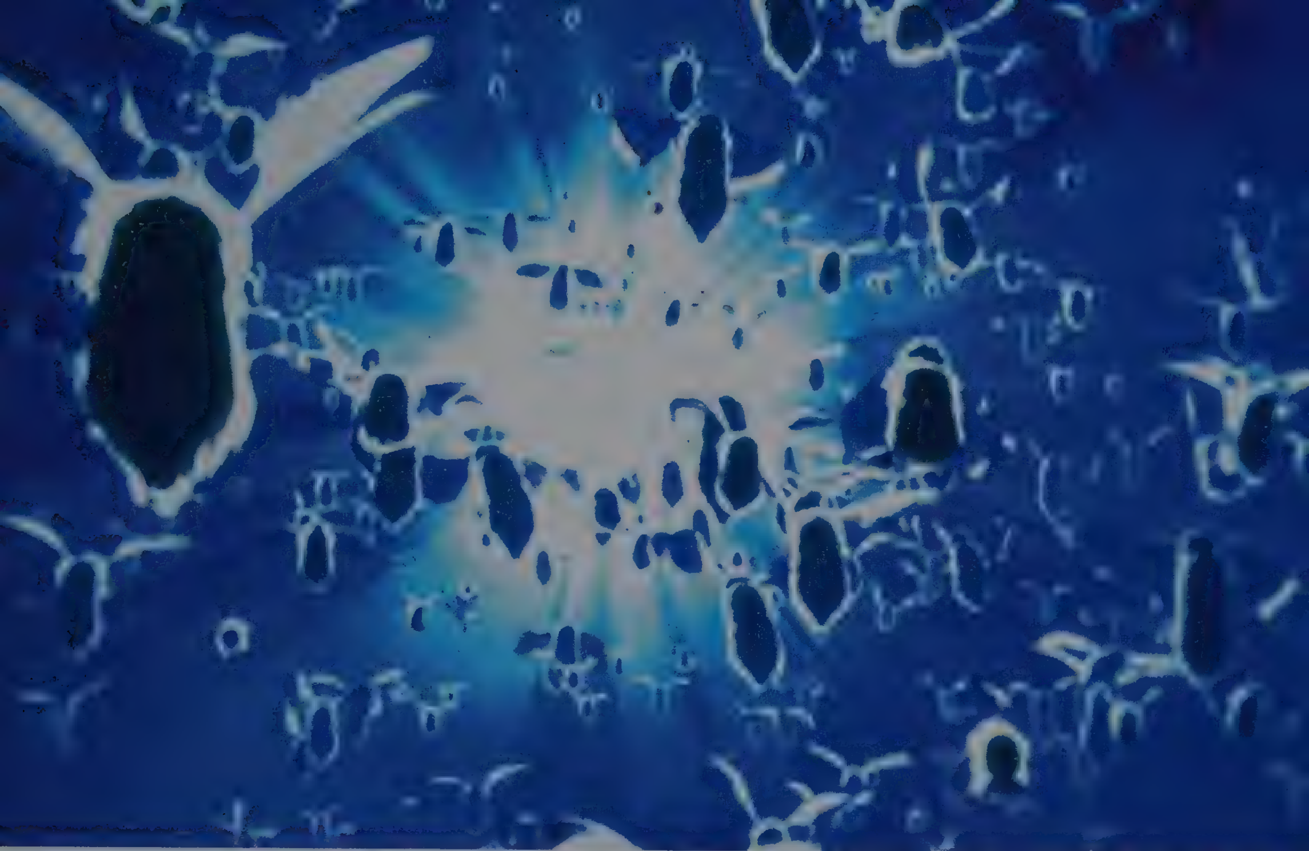
Effective lighting design starts with the visualization of the effect of lights in a specific environment. Fortunately, three-dimensional rendering programs are capable of actually simulating for us specific lighting arrangements. But even when using a computer rendering program to visualize light, any lighting designer is likely to achieve a greater degree of sophistication, beauty, and efficiency if he or she spends some time imagining visualizing the effects of the planned lights before trying them with the program.

An easy way to visualize lighting consists of starting with a dark space, turning the spotlights on and then adding the ambient light in small increments. By turning on the spotlights (or any other secondary light) first, you can focus on their lighting effect because much of the scene will still be quite dark. By turning on the ambient light (or the major point light) second, and in small increments, you can visualize the blending of secondary lights with the main light. This way the overall lighting effect shows through (or builds up) while retaining at all times the light accents. If necessary, those accents—usually in the form of spotlights or colored lights—can still be turned up or down after the ambient light has been defined, and adjusted to the requirements of the scene.

The lighting design can also be visualized by starting with a space that is already lit with ambient light. In this method the lighting accents are added toward the end of the process. While the final result can be the same whether one starts visualizing a dark room or a lighted room, I find that the latter requires more concentration, greater visualization power, and perhaps a little more lighting skill.

Animating Light

The position and attributes of light sources in a scene can be animated using the keyframe interpolation techniques described in Chapter 11. These techniques include the use of parameter curves and motion paths. A wide variety of lighting effects that affect the mood of a scene can also be created by animating the intensity of a light source as well as its color, cone angle, and fall-off. Moving lights in a three-dimensional environment, however, should be done

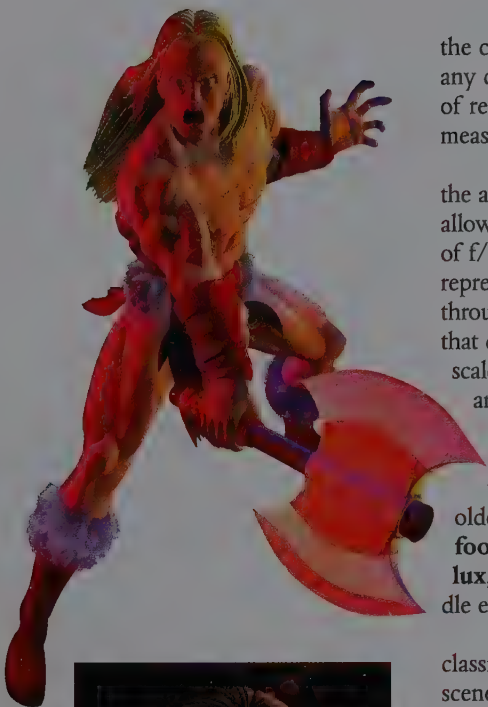


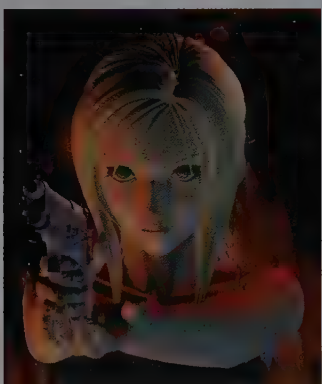
8.1.3 A key light placed in front of and facing the camera is used to represent the journey of the souls in one of the last shots from the film *Bunny.* The halos around the moths helps to make them look like ghostly angels. (© 1998 Blue Sky Studios.)

with restraint and a clear storytelling purpose because overly animated light sources can be a great source of visual distraction.

Measuring Light

The ability to control the aperture of a camera's virtual lens, for example, which is a cornerstone of everyday cinematography, is just starting to be implemented in computer animation software. Perhaps one reason behind this situation is the fact that in computer animation both the camera and the lights are synthesized from scratch, while in a live action film light has to be measured in relation to the characteristics of the film stock and the desired look. A small mistake in the measurement of lighting on location can impact many other aspects of live action production in adverse ways, while in a computer animated feature there is the opportunity to interactively fine-tune the lighting before committing it to film. An important exception to this situation happens when live action and computer animation overlap for the creation of a visual effect. In that case visual effects supervisors do take **light measurements on location** and the lighting technical directors do their best to replicate the lighting conditions with their computer tools. But some of the elements in the process of matching the live light with





8.1.4 (Top) The colored lighting on this Nordic warrior is as dramatic as his pose and gestures. (Bottom) Tinted lights were used to soften and color the shadows.(Top: © 1999 Midway Home Entertainment Inc. All rights reserved. Used by permission. Bottom: © 1998 Square Co., Ltd. All rights reserved. Based on the novel *parasite EVE* by Hideaki Sena, first published in Kadokawa Horror Bunko. Character designed by Tetsuya Nomura.)

the computer-generated light still include a lot of trial and error. In any case, while this situation changes it is worth mentioning a couple of relevant traditional lighting concepts and techniques related to measuring the intensity of light, and a scale to measure grayscales.

The **f/stop** is a unit to measure how much light passes through the aperture of a lens. In theory, an absolute aperture of the lens would allow all light passing through it to reach the film. The common scale of f/stops is roughly based on the square root of the number 2, and it represents a doubling of the additional amount of light that makes it through when the lens is opened, or the additional amount of light that does not reach the film when the lens is closed (Fig. 8.1.1). The scale of f/stops represents how traditional film reacts to light. There are a couple of additional units that are also useful to measure the intensity of light—the candela and the footcandle (Fig. 8.6.1).

The **candela** represents the amount of light emitted by a certain type of light, and it is a development of the standard candle, an older unit based on the amount of light created by a candle. The **footcandle**, and its metric system equivalent, the **metercandle** or **lux**, measures the amount of light that falls on a surface (one footcandle equals 10.764 lux).

The **zone system** is a technique widely used in photography to classify and balance the amount of light distributed throughout the scene. In essence the zone system, developed by photographer Ansel Adams, catalogs the gray levels in the image on a scale that goes from a pure black that barely reflects any light (Zone 0) to an absolute white (Zone X). The gray tones of each scale between Zone 0 and Zone X are separated by one f/stop (Fig. 8.1.2). **Zone V**, which is right in the middle of the scale, is of particular importance because at about 18% reflectance of light it represents the average reflectance of objects. The 18% gray tone is a reference widely used by VFX supervisors when measuring light on the set.

White Light

Most of us assume, incorrectly, that all natural light—and artificial light to a lesser extent—is white. But light, in fact, is rarely white. Light is usually tinted. Few elements in nature (perhaps water) are as chromatically dynamic as light (Fig. 8.1.3). The color of light changes with the time of day, the weather, the landscape, and the location on the planet. Just think of the chromatic differences, for example, between the light of a sunny winter afternoon in the Nordic fjords or a summer sunset in the stormy Caribbean or high noon in the clear spring skies of the Australian rocky desert. Nordic winter light might have a slight blue tint, while the light of a stormy Caribbean sunset might be the pink color of the *mamey* tropical fruit, and the spring Australian desert's light might have a slight yellow tint. The color differences in the three examples above might be subtle, but they are meaningful if one is trying to simulate environments like those with a three-dimensional rendering program.

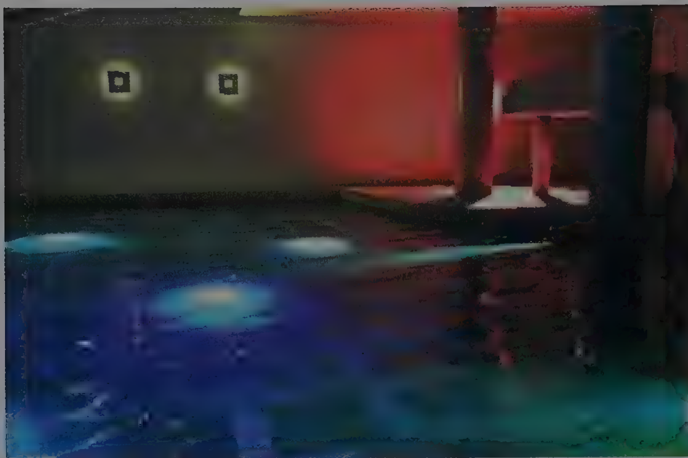
The lighting effect created by a **lightning** storm can be recreated by inserting one or two white frames in the sequence just a couple of seconds before the sound of thunder is heard. After that a very strong light—placed in the area near where the lightning is supposed to have fallen—is suddenly turned up to a bright white color, and then dimmed in a flickering way.

8.1.5 Tinted lights on the foreground elements were used in this shot to create a melancholic chromatic contrast between foreground and background. (*Dragon Hunters* © MMVII Futurikon Films, Trixter, LuxAnimation, France3 Cinéma, RTL-Tvi, in coproduction with Mac Guff Ligne.)

Colored Light

We can achieve startling lighting effects by using **colored lights**. The results are always reminiscent of the performing arts where performers are literally tracked around the stage with colored lights (Figs. 8.1.4 and 8.1.5), or dance clubs where much of the festive atmosphere and visual chatter is created with constant sequences and patterns of colored lights. The visual power of colored lights is so great, however, that they must be used with prudence, especially when lighting spaces or situations where a festive atmosphere would be distracting.

A pleasing visual surprise that is common in circus performance scenes happens when the projected lights of colored spotlights overlap with one another. This lighting resource owes its startling force to the unexpected colors that result from the mixture of colored lights (Fig. 8.1.8). Audiences are somewhat familiar with the results of mixing primary **pigment-based colors** with one another. Most have experienced this first-hand in elementary school or earlier: red and yellow makes orange, blue and yellow makes green, and red and



8.1.6 This image was created with a lighting model based on optical phenomena, such as the scattering and absorption of the light in the water. This model is capable of creating subtle details of the reflection and refraction of light on the water surface, the scattering and absorption of light in the water, and the shadows cast on the water surfaces. (Courtesy of Hideo Yamashita and the Computer Graphics Research Group of Hiroshima University.)

blue makes purple. Mixtures of **light-based colors** are startling because they follow the physical rules of light-based **additive color systems** as opposed to pigment-based **subtractive color systems**. It is always entertaining to puzzle your friends with a demonstration of the basic color mixtures in a light-based, three-dimensional rendering system: green and blue make cyan, blue and red make magenta, and (my favorite one) red and green make yellow (Figs. 8.3.2 and 8.6.1).

Tinted Light

Using tinted lights is a less dramatic but more subtle lighting effect than using colored lights. Using tinted lights is also a common technique in the lighting of simulated three-dimensional spaces—especially in determining a mood for the scene.

Using tinted lights can be an effective method for creating a cohesive atmosphere. **Tinted lights** create an effect similar to a coat of overpaint or varnish on the layers of paint, which unifies objects of disparate colors or surface finishes (Fig. 8.3.3). Tinted lights are created by selecting a slight coloration for the light emitted by the light source. When using the HSB color model to describe a tint, the saturation values should be low so that the color is washed out, the brightness values should be high so that the tint is not too dark, and the hue values could vary depending on the coloration desired for the tint. When using the RGB color model, each of the three values (red, green, and blue) would be high, so that the resulting color would be bright and not too saturated.

Light and Translucent Materials

The effect of light reflected off the surface of **moving water** can be recreated by placing spotlights shining up through a surface that represents water and that has an animated shape (Fig. 8.1.6). Some rendering programs have the ability to accurately simulate the effects seen when light travels through a volume of water (Fig. 8.1.7). These light patterns, called **caustics**, are caused when multiple focused rays of light are reflected or refracted onto one another.

8.1.7 (Opposite page, top) These underwater scenes display some of the caustics effects of light being refracted by the volume of seawater. (Detail of *Invisible Ocean*, a film by François Garnier. Executive Production: Ex Machina. © 1998 Monaco Inter Expo. Special thanks to the Oceanographical Museum of Monaco. Images courtesy of Ex Machina.)

8.2 Types of Light Sources

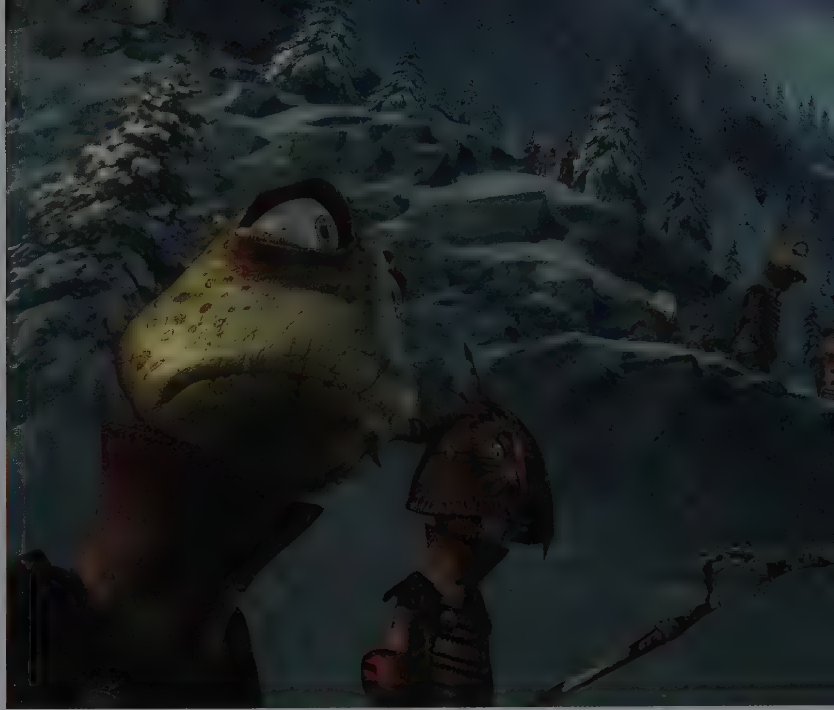
There are several basic types of computer-generated light sources depending on the way they irradiate light. Simulated light sources

generally include point lights, spotlights, linear lights, area lights, infinite lights, and ambient lights (Fig. 8.2.1). Most rendering programs automatically create one or several **default lights** in the three-dimensional scene. Default lighting schemes can usually be customized and may consist of an ambient light source, an infinite light that simulates the intensity and position of the sun, or a point light that is placed above and behind the camera or in any other XYZ position. Some rendering programs do not provide default lights, which means that if it is rendered without any new lights, the result will be like looking into a windowless room without any lights. As you practice creating computer-generated lights, try to learn as much as you can about the major types of traditional lights and their characteristics (Fig. 8.6.1).

8.1.8 Scene from the *Egg Cola* trailer where colors were added with light, which resembles the traditional painting approach. Between 20 and 25 Maya's dome and sphere light sources were used for the control room, and sphere lights only for the background. Characters were modeled with NURBS. (© Independence, Inc.)

Point Light

A **point light** casts light evenly in all directions. For this reason a point light is also called an **omnidirectional** light (literally "in all directions"). Point lights are the simplest type of light source, and they can be placed anywhere in the scene. Point lights can be placed, for example, outside of the field of vision of the camera, behind an object in the scene, or even inside of objects. The effects of point lights placed inside of objects varies between software programs, but in many cases the light will shine through the walls of a transparent



8.1.9 Colored lights are effective tools to set the mood in these *Oddworld: Stranger's Wrath* scenes. (© Oddworld Inhabitants.)

object as in the case of a lightbulb. An incandescent lightbulb is a simple example of a point light. A star, a candle, and a firefly are also point lights but require additional effects.

Spotlight

A simulated **spotlight** is like a point light to which "barn doors" of the type commonly used in the performing arts have been added. Spotlights cast light in a cone shape and only in one specific direction. Spotlights have some unique characteristics: a variable-angle cone of light, and a light fall-off factor (Fig. 8.3.5). Flashlights, lamps with shades, jack-o-lanterns, and the light reflectors used in stage or movie productions are all examples of spotlights (Fig. 8.2.2).

Spotlights that are dimmed or turned up produce an effective way of attracting the attention of the audience to a specific area or situation in a three-dimensional scene. A narrow soft-edged spotlight can be especially effective for highlighting the action when the **illumination level** in the scene is low (Fig. 9.4.5). A spotlight in a dark scene can add a feeling of suspense or fear to the shot because the lighting effect might remind the audience of a search for something—or someone—who is hiding, or trying to hide from someone—or something—that is looking for us.

Infinite Light

Infinite lights are so far from the elements in the scene that their light rays reach the scene parallel to each other. **Infinite lights** are

also called **directional lights**, and they behave like stars in the sky. But unlike stars, computer-simulated infinite lights can be placed anywhere in the environment, are massless, and their intensity can be modulated (Fig. 8.2.3). In many programs infinite lights have a constant intensity, and do not decay as they travel through space. The **sun** is a special case of an infinite light source that can be accurately placed above the scene by typing the latitude and longitude of the location plus the exact time of day and date when the simulated scene is taking place (Fig. 8.5.1).

8.1.10 Even in a quick reaction shot good lighting helps to delineate the subject and contribute to the mood of this scene from *A Gentlemen's Duel*. Notice the sharp lighting in the white gloves and the rim lighting on the face. (Created by Blur Studio, Inc.)

Area Light

Some programs provide **area lights** in the form of multiple lights grouped together, or a single large area of light (Fig. 8.5.3). Area lights can be scaled to almost any size but are more efficient when kept small, and are usually rectangular or round in shape (Fig. 9.4.5) Area lights are especially useful for lighting small areas uniformly like the way, for example, custom jewelry is professionally photographed by being placed on a translucent light box or between two light boxes. Area lights can also be used to simulate the reflection of light coming into an interior space through open windows.

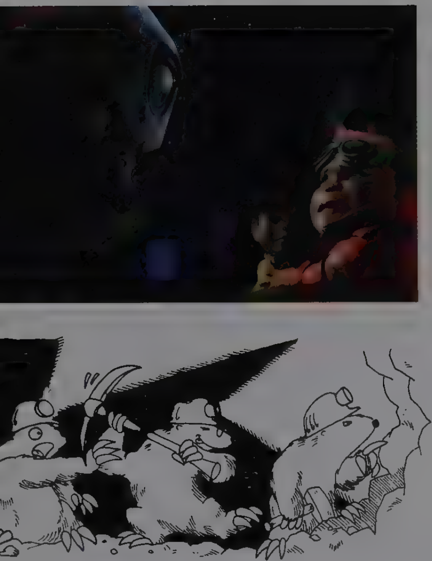
Common Types of Computer-Generated Light Sources
Point Light
Spotlight
Infinite Light
Area Light
Linear Light
Ambient Light

8.2.1 The basic types of computer-generated lights are defined by the characteristics of the source, such as the direction and angle of the beam, their shape and placement in space.

Ambient Light

The light radiated by the **ambient light** source is distributed evenly throughout the entire scene. The term ambient light is often used generically by different software programs, and technically speaking,

it does not always refer to an ambient light source. In some cases it refers to a point light source that is created automatically by the program for each scene. Even though an ambient light source can be placed in a specific XYZ position in three-dimensional space, it is best to think of an ambient light as coming from all directions (Fig. 8.2.4). The ambient light source often determines the general **level of illumination**, or shade, of a scene and almost always there is only one ambient light source per scene.



8.2.2 Spotlights are effective for highlighting the action in the scene. They sometimes have a visible cone of light and a visible light fall-off factor. (Top, © 2007 SOFA Studio, Ltd.)

Linear Light

The light of the fluorescent tubes used to light so many public spaces can be simulated with linear lights (Fig. 8.2.5). **Linear lights** have length but no width, and they can also be scaled to any size. Using linear light sources should be exercised with care because their computation in some cases can be much more time-consuming than the combination of several point lights.

8.3 Basic Components of a Light Source

The main elements of all simulated light sources include position, color and intensity, decay and fall-off, glow, and shadows (Fig. 8.3.1). In addition, spotlights are also defined by their orientation and cone angle. Lighting software makes it possible to edit each of the individual components of a light source separately. A few light attributes can also be saved in a file, called a **light shader**.

Position and Orientation

Both the **position** and **orientation** of a light source can be controlled with the standard navigation or geometric transformation tools provided by all rendering programs: simple and combined translations and rotations. In the wireframe display mode, light sources are usually represented with a variety of graphic symbols; for example, a lightbulb for a point light, a lantern for a spotlight, a sphere attached to a straight line for an infinite light, and so on. But when a scene is rendered, the actual light sources themselves (not the light coming from them) can usually be seen, unless they are made into **invisible lights,** in which case they do not appear in the final rendering. In many programs, the light sources are visible by default, and when rendered they appear in the image as bright spots or as small three-dimensional objects that look like the graphic symbols commonly used to represent the light source in the wireframe mode.

8.2.3 The sun is a good example of an infinite light.

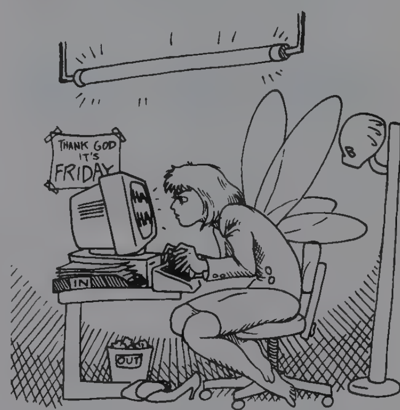
Color and Intensity

Simulated light can have virtually any **color**. In most rendering programs, the color of lights is usually specified using a light-based or **additive color model**. The RGB (Red, Green, Blue) model and the

HSB (Hue, Saturation, Brightness) model are both additive color models (described in Chapter 6). Most programs provide both color models to work with. In the RGB color model, a color can be specified by its individual red, green, and blue components. The numerical ranges used to specify color vary from program to program. They can range, for example, from 0.000 to 1.000, 0 to 255, or 0 to 65,535 depending on the color resolution and precision of the system. Unlike pigment-based color models, where the color mixture gets darker as more color is added, in the RGB color model the color of the light will become lighter as the amount of color mixed increases (Fig. 8.3.2). Note that in a three-dimensional environment, the color assigned to objects is always influenced by the color of the light sources as well as the position of the object in relation to them (Fig. 8.3.2).

When using the HSB color system it is possible to specify the intensity of a light source independently from its color or hue. For this reason it is easier for most people to quickly define the color of lights with this color model than when using the RGB model. One of several tools for selecting colors visually within the context of the HSB color system is illustrated in Figure 8.3.4. Most three-dimensional rendering programs provide dimmers to control the **intensity** or **brightness** of a light source. Intensity values commonly range from 0.000 to 1.000 with maximum intensity represented by a number one and minimum intensity (or OFF) being represented by zero when using the HSB color model (Fig. 8.3.11). Some programs offer simple tools for boosting the intensity of a light source (Fig. 8.3.5).

8.2.4 The ambient light used to illuminate this scene from *Gopher Broke* creates a happy natural feeling. (Created by Blur Studio, Inc.)



8.2.5 The fluorescent tube is the classic example of a linear light.

Basic Components of a Computer Generated Light
Position and Orientation
Color and Intensity
Decay and Fall-Off
Beam Angle
Glow
Shadows

8.3.1 A list with the basic components of standard light sources simulated with three-dimensional computer animation software.

[illegible]		
Red	255-0-0	1-0-0
Green	0-255-0	0-1-0
Blue	0-0-255	0-0-1
Aqua	161-255-238	.631-1-.933
Cream	252-255-103	.988-1-.403
Rust	141-43-17	.552-.168-.066

8.3.2 Two columns with RGB numerical values expressing color in two different ways, using a scale of 0–255 and a scale of 0–1.

The intensity of the light source can be controlled independently of its color, but since the intensity and the color of light influence each other, almost any change in the color of a light affects its intensity. For example, if we have two red lights with the same intensity but one of them has a dark red color and the other a light red color, the latter will appear to be a light with a higher intensity.

Decay and Fall-Off

The **decay** value of light controls the force of a light source and, as a result, how far from the light source the light travels. A weak light decays rapidly, while a strong light decays slowly and travels far. In the real world the decay of light is always linked to the intensity of the light source that created the light, but in computer-simulated lighting decay it is often independent of the intensity parameter.

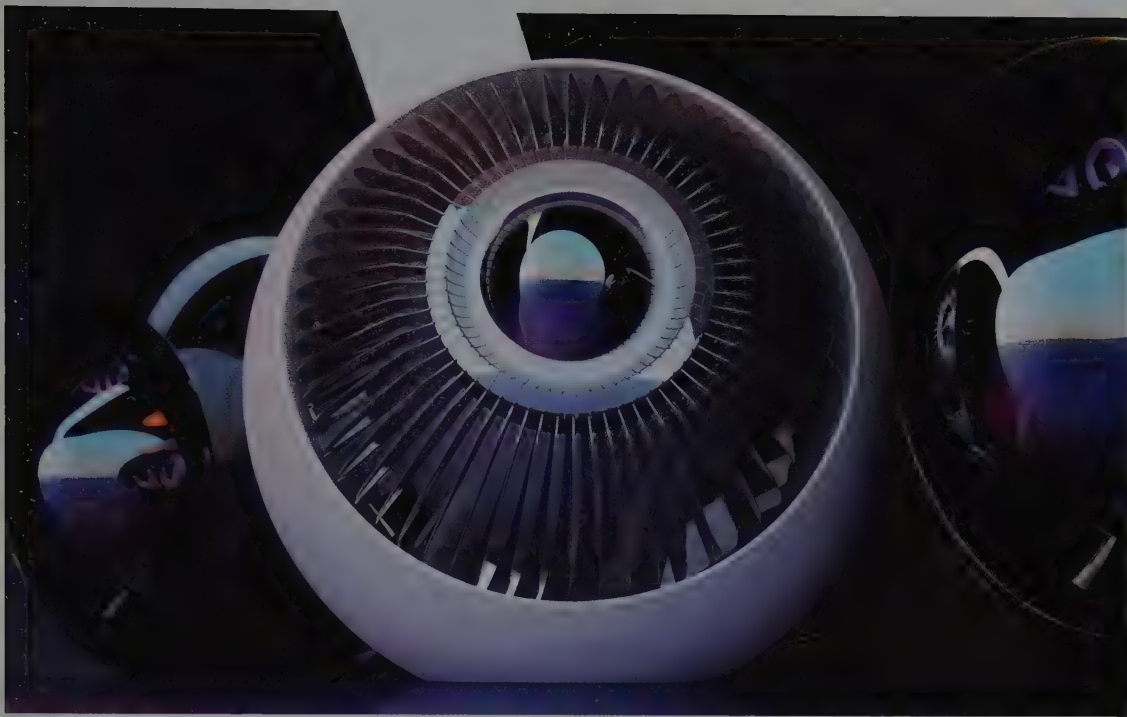
In most programs the decay parameter defines the force of the light—regardless of its type—as it travels away from the light source. The light created from point lights decays equally in all directions. The light created from spotlights, however, decays as it moves away from the light source, but also as it moves from the center of the beam cone toward the edges. This type of decay is sometimes called **fall-off**. Decay and fall-off can be controlled with linear interpolation for slow fading effects or with exponential interpolation for abrupt fading. The sharpness or softness of the edges of spotlight beams are controlled with the fall-off value.

Beam Angle

The **beam angle** feature of lights is a unique characteristic of spotlights. The cone angle of a spotlight defines the diameter of the beam of light and also the surface area covered by the light. This parameter simulates the barn doors in real spotlight lamps that control the **spread** of the light beam (Fig. 8.3.5).

Glow and Cone of Light

It is possible with some programs to simulate a variety of glowing lights. The **glow** of a light is a circle of light that forms around the light source because the light is refracted and reflected by particles in the environment—generally ice, dust, or smoke. In some instances the glow of light is calculated based on the **bleeding** displayed by very bright light sources, instead of the refraction of light in a three-dimensional environment. The light bleeding effect is common in situations when a photographic camera is pointed directly at a light source, and the resulting photograph has a bright spot with light bleeding around it. The difference between these two methods of creating a glow in computer-generated lights is that one (refraction) is based on three-dimensional calculations, while the other (bleeding) is based on two-dimensional calculations (Figs. 14.1.4 and 14.2.7).



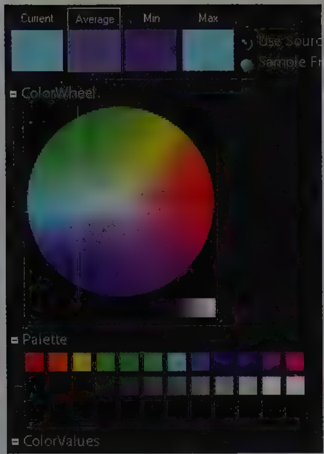
The glow of a point light usually occurs as a circle or **halo** around the light source. The glow of a spotlight occurs in the form of a **cone of light**. Circular and conical light glows are often called **volumetric lights**, and they are both defined by the decay of the light source. Linear decay results in a gradual fading of the glow effect, while exponential decay results in a sudden vanishing of the glow. Conical light glow is further controlled by the spread or beam angle of the light source. The thickness or frequency of the particles in the environment that cause the light glow is controlled with parameters that simulate the size, orientation, motion, and opacity of the particles in the environment that cause the light glow (Fig. 8.3.6).

Lens flare is an effect that is related to light glow, and it simulates the refraction of light inside of a camera lens. Lens flare creates the rings or stars caused by the refraction of light inside camera lenses, and it is also a two-dimensional effect commonly available in many postprocessing and compositing programs.

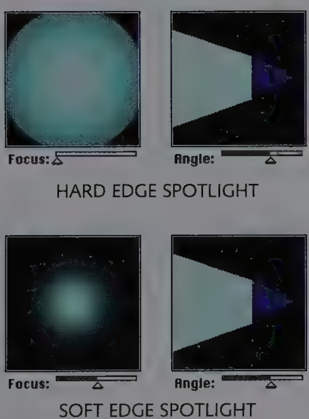
8.3.3 The material shader in these objects has a base color surface (black and white, respectively), and an additional glossy layer that creates a richer look of surface "depth." A single Sun-like ambient light illuminates the scene where the objects are placed within two concentric domes. The black inner dome has a slit that allows the outer black dome to show through. An additional sphere emitting orange light was suspended inside the dome to create an expressive tinted light accent. The simple objects are inspired by jet turbine engines. (Image courtesy of Toru Kosaka.)

Global and Local Lights

Global light sources shine on all objects in the scene that are directly exposed to the light sources. The light sources in a scene are global by default. Rendering methods like radiosity are used when even the three-dimensional surfaces that are not directly exposed to a light source receive some light in the form of penumbra or **diffuse**



8.3.4 The Shake Color Picker simplifies the visual selection of color within the ranges of the HSB (Hue, Saturation, Brightness) color model. Color can also be selected by inputting numerical values in HSB or RGB (Red, Green, Blue) formats. (Courtesy of Apple Computer.)



8.3.5 These two pairs of images compare two spotlight beams with the same cone angle values, but the one on top has sharp edges with little fall-off, and the lower one has soft edges with fall-off. (Graphs from Infini-D 2.6. © Specular International, Ltd.)

interreflections. The illumination effect of global lights is largely dependent on their position and orientation in the scene and their brightness, but objects directly exposed to global lights always reflect some of that light.

A different situation occurs with **local light sources**, also called **linked** or **selective light sources**. A local light source sheds its light on the objects linked to it, and this link can be exclusive or inclusive. An **exclusive link** between the light source and the objects limits the light projected by a local light source to fall only on the objects linked to it. In some programs an exclusive link may override the fact that a linked surface may not be exposed directly to the light source. This is as if the light source could magically travel through opaque objects that would ordinarily block light from reaching the linked object without affecting them. An **inclusive link** allows a local light source to always illuminate the objects linked to it as well as other objects in the scene that may be directly exposed to it.

Establishing links between light sources and three-dimensional surfaces can be an effective way to achieve complex lighting situations, but it can also increase the management complexity of a scene. Local light sources are implemented in different ways by different programs. For this reason local light sources should be used with restraint. From the point of view of an object that is linked to one or several local light sources, the object will be illuminated only by them and not by any other light sources that may be active in the scene.

Shadows

In principle, all light sources cast **shadows**. But computer-generated shadow-casting is a feature of lights that can be turned on or off. Since shadow-casting is also an optional attribute of objects and shading techniques, the final visual appearance of shadows is determined not only by the attributes of the shadow but also by the attributes of the shadow-casting object and the rendering method employed. Shadows can be defined by several parameters, including color of the shadow, color of the penumbra, and softness of the shadow edge (Figs. 8.3.7–8.3.9, 8.5.5, and 8.6.6).

The portion of a shadow that blocks direct light altogether is called **umbra**. It is the inner part of the shadow. The area in the edges of the shadow that blends with other lights in the environment is called **penumbra**. The **softness** of the edge of a shadow is controlled in a variety of ways. With many rendering methods—excluding ray tracing—the soft edges of a shadow can be controlled by the distance between the light source and the shadow-projecting object. The shadow edges will be sharp as the light source moves farther away from the object. The number of levels of shadow tracers influences the softness of a shadow when ray tracing is used. When using radiosity-based rendering, the shadow edges are soft when the surfaces in the environment create a lot of diffuse inter-reflections (Figs. 7.3.2, lower right, and 10.1.1).

Patterns of shadows are often used to create a mood in the scene. A common way to create these patterns is by projecting light through cutout stencils, also known as **gobo lights**.

8.3.6 (Top) The path of light is revealed by particles in the space. (Bottom) Reflectors with glow pointing to the sky in this architectural visualization of the Samsung headquarters building in Seoul, Korea. (Top: *Même les pigeons vont au paradis*. Bottom: © BUF, Director Samuel Torneux. Bottom: Courtesy of Rafael Viñoly Associates.)

8.4 Lighting the Scene

Those who realize the importance of lighting can appreciate the importance of a systematic approach to lighting. Without light, the entire contents of the world could not be appreciated visually. Without adequate lighting, shapes, colors, and textures can only be experienced halfway. Think, for example, of some of the elements of a beautiful face: the features, the proportions of the shapes and their curvatures, the evocative color of the eyes, the subtle coloration and the texture of the skin, and the weight and flow of the hair. A successful lighting arrangement can reveal all of these elements and present them in a harmonic way. But a poorly designed lighting scheme will fail to bring up the full experience and depth of the appealing face. Strong shadows in the eyes, for example, may obscure or hide the eye color. Unbalanced shadows around the nose might distort its delicate balance. Lighting the face from certain angles might flatten the jawline or dilute the seductive meaning hidden in the shape of the lips.

Next we will review the basic styles of light sources and the areas in the scene that require lighting. We borrow concepts from traditional portrait photography and stage lighting. The latter can be easily adapted to the lighting of computer-simulated environments because in both situations the scene is totally dark unless we turn the lights on. The areas in the scene include the main action area, the secondary action areas, and the backgrounds. The basic styles of light sources include key lights, fill lights, kicker lights, and practical lights.

8.3.7 *The Cathedral* tells the story of a pilgrim who arrives at a magical place on the edge of the world. The lighting contributes to the staging of the action and the suspense of the storyline. (© 2002 Tomek Baginski and Platige Image.)

Main Action Area

The **main action area** is the area in the scene where most of the action takes place. It may consist of a small area—for example, if a molecular interaction is being rendered, or a large area in the case of a car chase. In computer-generated scenes the main action area might be located in a quiet indoor space or extend over firewalls and through colored rain. Depending on the mood a couple of spotlights might be enough for a simple shot of a dialogue scene between two characters, but several point lights and spotlights might be needed to delineate the motion of a dozen fantastic characters dancing back and forth on the stage. Often times lighting arrangement in this area require several variations when the scene is shot from different points of view. This would be done to emphasize different aspects of the subjects to cameras placed in different locations. In most situations the lighting of the main action area defines the overall mood of the scene. For that reason the light sources used to light the main action area on a theater stage are often called key lights and are often used in conjuction with the fill lights to set the lighting tone. In traditional stage lighting design it is not uncommon to divide the main action area into several sections, depending on the action that is to take place, and to assign each section a certain number of lights—for example, between two and five spotlights per section (Fig. 8.4.1).

Secondary Action Area

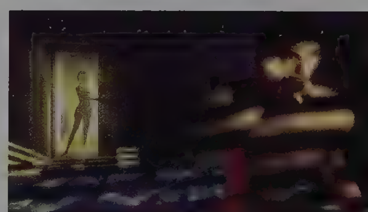
The **secondary action area** is the place in the scene into which some of the action eventually spills. For example, two characters in a

8.3.8 The sunlight floods this shot that evokes the simple pleasures of lunching in the backyard. In spite of their well-defined edges, the inside value of the shadows is quite light. By simply making the shadows a darker value this *Polygon Family* scene would have a different connotation. (© POLYGON PICTURES/IPA/NK-EXA.)

scene that takes place in a living room spend most of their time sitting on the couch (main action area). But at some point one of the characters gets up and walks to the bookshelf (secondary action area), picks a book, and returns to the sofa. The lights illuminating the bookshelf and the book may be on at all times throughout the scene, or only turned on for the shot with the character walking to the bookshelf. The number of lights needed to illuminate the secondary action area in a small environment is usually smaller than the number of lights in the main action area. Secondary action areas in exterior shots might require dozens of light sources.

Background

Theater stage backgrounds are traditionally fabricated with large cutouts and painted flat elements, with a prop here and there. Most theater stages use a flat **background** because of the limited area of the platform used by the actors to perform. Some sets designed for computer animation do use backgrounds: horizontal or vertical flat planes with texture maps—of bricks, for example—or convex surfaces with procedural maps of animated clouds or even a composited photograph. Backdrops with texture maps are sensitive to colored light. Minor chromatic changes in the lights that illuminate the backgrounds oftentimes have a significant effect. Generally the stage and the background in computer-generated environments are closely related, and in many cases share geometry. Some computer-animated scenes take place in enclosed spaces that resemble the **sound stages** in which movies are filmed; others take place in interiors that resemble an intimate theater **stage**; others take place in open spaces that



8.3.9 The character in the background is turned into a shadow by using lighting from behind. The venetian blind shadow patterns on the foreground character are created with gobo lights. (Images created by Mondo Media, CA. © 1999 Pulse Interactive.)

8.3.10 The lighting is simple but interesting in this shot from *Polygon Family* showing a businessman waiting for the train to return home. Notice the presents for his kids, and how the light shadows add warmth to the moment. (© POLYGON PICTURES/IPA/NK-EXA.)

remind us of live action movies being shot **on location**, with significant scenery and surroundings.

Three-Point and Four-Point Lighting

Lighting subjects in traditional portrait photography is commonly done with three or four lights. Both setups are effective for creating good definition and pleasant modeling of the subject's features while avoiding distracting highlights and harsh shadows. **Three-point lighting** always includes a key and a fill light, and a rim light or a kick light depending on style, preference, and desired effect. **Four-point lighting** includes the three lights just mentioned plus a background light. The dominant lights in both setups are the key and the fill lights, and there are multiple stylistic approaches on how to balance the lighting ratio between the two (Fig. 6.1.2).

8.3.11 (Opposite page, below) The contrast between the sharp light reflections and the fill lights emphasizes the dramatism of these scenes from *Kaena*. (© 2003 Xilam Films–StudioCanal–Group TVA Inc.)

Key Lights

The **key light** is the dominant light on the subject or point of interest and it is also the one that sets the overall mood in a scene. Just a single key is required when lighting a still subject, but several key

lights may be needed in a shot with a moving subject or a shot with multiple subjects moving throughout the scene. Key lights are usually positioned above the subject and at an angle, and they delineate and model the shape of the subject (Fig. 8.6.6). Shadows projected by the key lights can be softened by the fill lights.

Fill Lights

The **fill light** is also known as an ambient light, and its main purpose is to soften the key light. A fill is usually positioned at the subject's face level and on the opposite side of the subject as the key. In configurations with multiple lights the fill lights help to define the overall color tone of the entire scene, as well as blend multiple keys in the scene. Fill lights do not project shadows and they can be soft and tinted or quite bright. Depending on the lighting effect desired computer-generated fills can be created with spotlights or with directionless infinite light sources. The **eyelight** is a specialized version of a fill that is used to draw a twinkle in actors' eyes. A **bounce light** is variant of a fill where the light is projected onto the subject by bouncing it off a reflective surface.

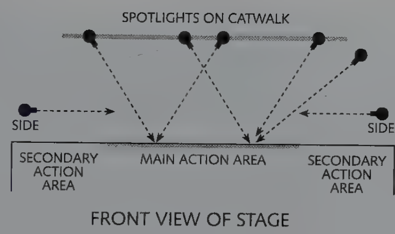
Kick Light and Rim Light

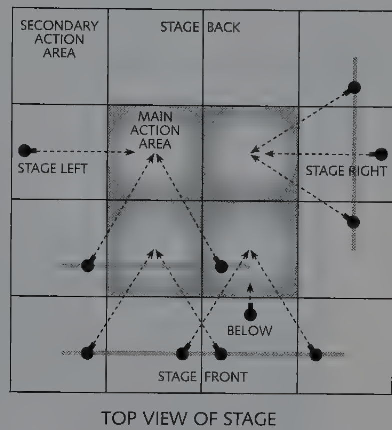
Both the kick and rim lights are used to accentuate and define the shape and volume of the subject—usually the face or head—or an object. The **kick light** is usually positioned to the side of the subject and above. A **kicker**, as a kick light is also known, helps to separate the subject from the background by adding a small amount of sharp light to the front or the side of the subject. A **rim light**, also called a **back light**, is used to sharpen the silhouette of the subject by creating an outer edge of light around some of it (Fig. 4.5.6).

Practical Lights

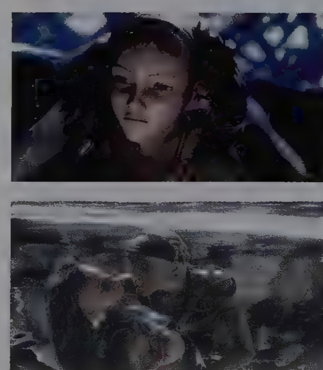
On occasion the actual light sources are shown in the scene, for the viewers to see as part of the composition. These **visible light sources** might include candles, fireplaces, flashlights, lamps, televisions, open refrigerators, and comets (Figs. 8.3.7, 8.4.2, 8.4.3, 8.5.6, and 8.6.6). These light sources are called **practical lights** in traditional film and video production, and when used they often play an important storytelling role—for example, when an important character moves through darkness with the help of a practical light, trying to accomplish something that will bring irreversible change to the story. Computer-simulated visible light sources can be simulated with point lights, and their effect can be accentuated by rendering them in conjunction with particles, volumetric lights, or by simply placing them inside of a shape that is rendered with translucency and glow.

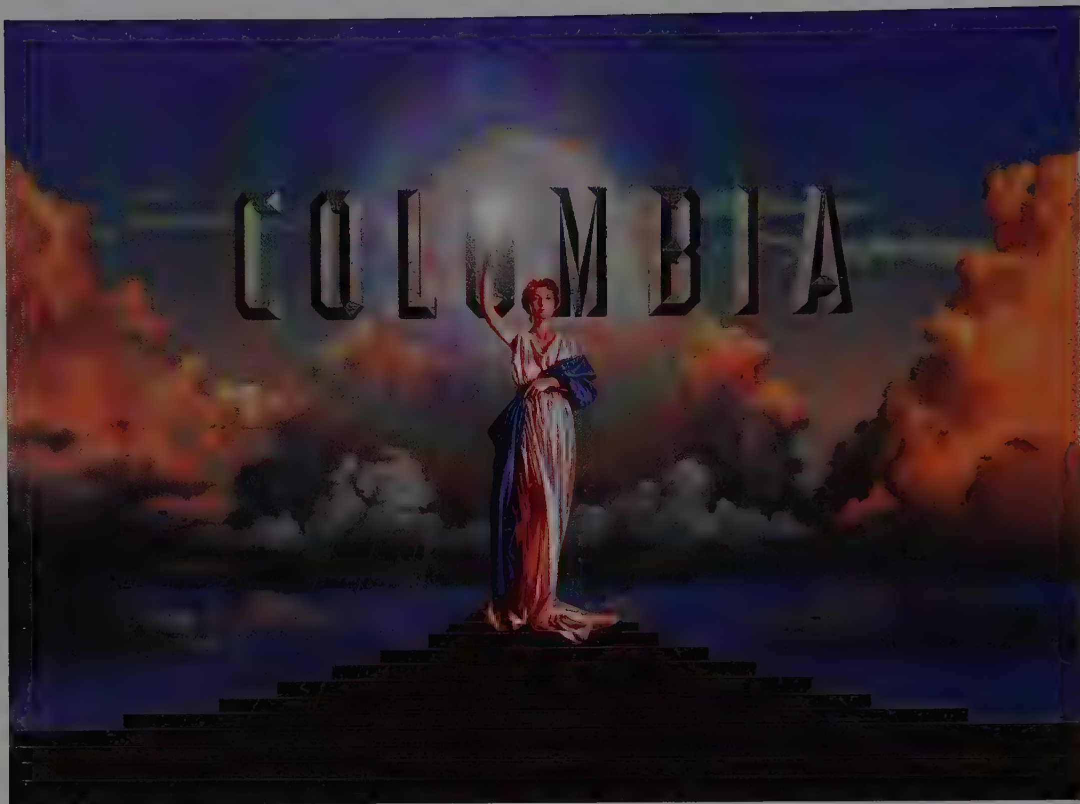
The sun or the moon, for example, are commonly represented as light sources. They can be easily defined as point lights with





8.4.1 This diagram shows a lighting layout of the main and secondary action areas. In traditional stage lighting the size of each grid subdivision corresponds to an area that can be filled by a spotlight beam.





8.4.2 The practical light held by this icon of filmmaking dominates this image that also includes a significant amount of digital painting and compositing techniques. (Produced by Kleiser-Walczak Construction Co. for First Light, Inc. and Columbia Pictures. Animated by Jeff Kleiser, Diana Walczak, and Ed Kramer. Courtesy of Kleiser-Walczak Construction Co.)

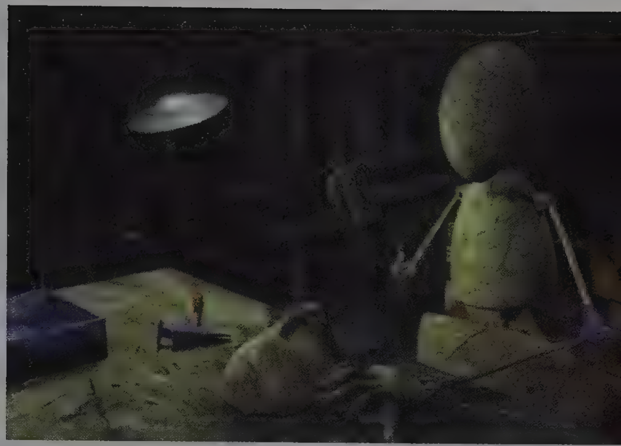
medium to high intensity and little or no fall-off. The color of these lights may have a warm tint in the case of the sun, for example, or a slightly cool tint in the case of the moon. The occasional flickering light of celestial bodies can be recreated by animating the intensity or the glow of a point light source.

Special Lighting Effects

Practical lights with motion are often used to create **special lighting effects** that can be used to emphasize certain dramatic moments or to impose dominant moods. Fireworks, explosions, haze and fog, and lightning are all examples of special lighting effects (Fig. 8.4.4). Many lighting effects encountered in nature can be simulated with combinations of **moving lights**. Some of these lighting effects include the light emitted by lightning, fire, natural explosions like a volcano, and the light reflected off the surface of moving water and refracted through moving water like a waterfall.

Fire lighting effects that are off-screen can be achieved with a group of point lights and spotlights placed within an area that corresponds to the dimensions of the fire being recreated. The intense and constant lighting motion generated by a fire can be achieved with

wiggly parameter curves for position, cone angle, and each of the RGB colors in order to achieve maximum irregularity. The lighting effects created by light traveling through colored glass—for example, the effect created in an interior space by exterior lights traveling through stained glass—require a translucent image map, or a rendering technique like radiosity or ray tracing that calculates the effects of light traveling through transparent or translucent surfaces. See Chapter 11 for more about animating of light sources.



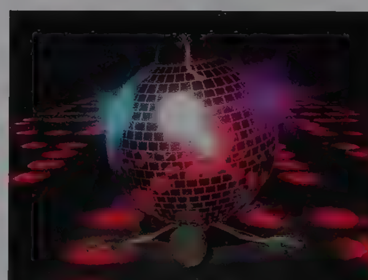
8.4.3 The light from the lamp (a practical light source) carries the lighting in this shot, while the flood light is kept very low. This lighting adds intimacy and mystery to this moment when a metaphysical character searches for his identity by trying on different masks. *Masks* won the first SIGGRAPH Electronic Theatre Jury Honors Award in 1999. (Image courtesy of Piotr Karwas, writer/director/animator/designer.)

8.5 Basic Positions of Light Sources

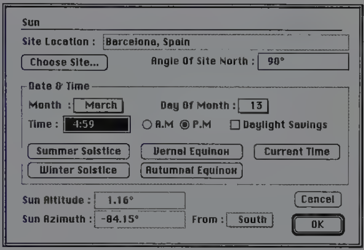
Computer-simulated lighting has the great advantage that lights can be moved around without having to worry about clamping them to spotlight bars or poles, as is the case in traditional lighting. Computer-simulated lights have the ability to float in space. Most three-dimensional rendering programs offer a choice between standard XYZ notation or spherical coordinates for positioning lights in three-dimensional space. The spherical coordinate system, as mentioned in Chapter 3, specifies the position of objects in three-dimensional space in terms of their altitudinal and azimuthal angles (above and around) in relation to a center of reference. The position of the sun, for example, can be described in terms of its altitude and its azimuth (Fig. 3.4.8). The **altitude** is defined by the angle of the light in relation to the horizon. The **azimuth** is defined by projecting the angle of the sun onto the east-west axis. This technique is especially convenient in architectural projects where the position of the sun has to be defined for calculating both the amount of shadow cast by a building and its surroundings, and the amount of direct sunlight received by the structure at any time of the day (Fig. 8.5.2).

Once light sources are positioned they can be aimed at specific objects or areas in the environment in a variety of ways. Centers of interest can be specified **numerically** by typing XYZ values, **visually** by pointing the **light vector** displayed by some light sources at the object in question, or **procedurally** by choosing commands (provided by some programs) that will automatically point one object—usually a light source or a camera—to another.

In principle, there is no limit to the **number of light sources** that can be placed in a three-dimensional scene. The only limitations are of a practical nature. The budget and timetable of some projects, for example, may determine the number of light sources and the style of lighting. Both in computer-simulated lighting as in real lighting, to create, place, fine-tune, and render lights requires time and money. Lighting requirements—whether real or simulated—have a wide range of complexity. Compare, for example, the lighting requirements of an indoor large-scale sports event to those of the



8.4.4 The disco ball, an example of moving lights, plays a heavy role in the conclusion of *Alien Song*. (© 1999 Victor Navone.)



8.5.1 The position of lights—especially the sun or other stars—can also be specified using the spherical coordinate system. This dialog box provides an easy way to define the position of the sun in relation to Earth anywhere and anytime. (form•Z dialog box. © 1991-1995 auto•des•sys, Inc.)

Basic Positions of a Light Source

Pair at 45 Degrees
Front from below
Front at subject's level
Side at subject's level
Side above
Side above and behind
Behind above
Behind side
Overhead

8.5.2 The basic positions of light sources are listed on top, and illustrated on the opposite page as they focus on a sphere.

close-up photography of a diamond ring. Both lighting situations require unique solutions.

Lights can be placed in a variety of places in relation to the camera and the subject. Five basic positions of light sources and their corresponding variations as they focus on the subject are examined here: a pair of spotlights at a 45-degree angle, front (below and subject's level), side (subject's level, above, and above and behind), back (above and side), and top. Figure 8.5.2 provides a visual summary of these basic lighting positions using only spotlights to accentuate each lighting effect. Different situations require different lighting arrangements.

45-Degree Pair

One of the most common lighting arrangements (in fact, it is usually called **ordinary lighting** by stage lighting designers) consists of two spotlights placed above, in front, and to the sides of the subject. In this common lighting configuration the lights are both focused on the subject at a 90-degree angle in relation to each other. Both lights are rotated 45 degrees around the vertical and horizontal axes. The **45-degree angle spotlight pair** represents a simple and effective way to have a generous amount of light that reveals the features of the subject as well as some detail in the form of shadows (Fig. 8.5.3).

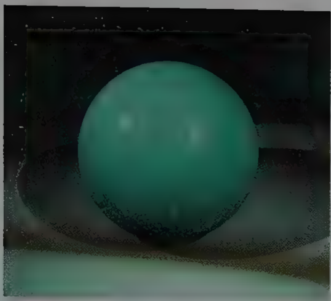
Frontal Light

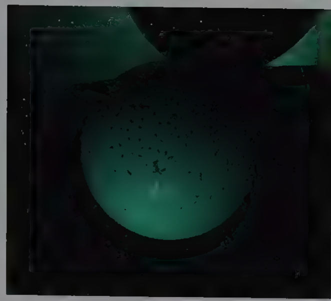
Frontal light from below is very effective for casting pronounced shadows both on the subject and the environment. Since we rarely encounter this type of light in natural surroundings, frontal lights from below can look artificial or overly dramatic. But they can also be quite effective for accentuating truly dramatic, scary, or otherworldly moments (Fig. 8.5.4). **Frontal light at the subject's level** tends to flatten the subject because it usually eliminates most of the deepest shadows, but it can also be used as a low intensity fill light for blending other spotlights in the scene.

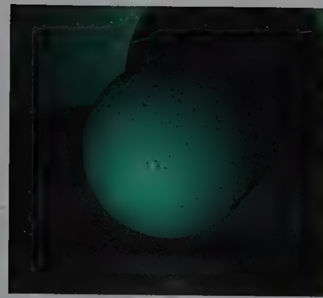
Lateral, Top, and Back Lights

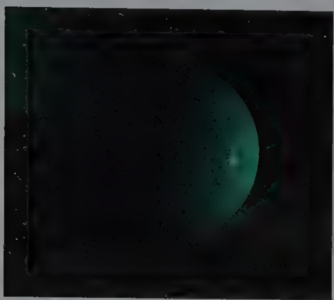
A **lateral light at the subject's level** is useful for increasing the contrast between light and dark. The accentuated shadows created with lateral light can have a powerful dramatic effect and add a lot of depth to the scene. Lateral lights should be used with care because they can easily overpower other more delicate lights in the scene (Fig. 8.5.5).

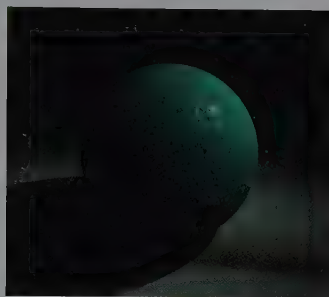
A **lateral light from above**—especially when used in pairs, one on the left and one on the right—creates an effect that is similar to the 45-degree spotlight pair described earlier but with slightly more pronounced shadows. One advantage of creating pronounced shadows with lateral lights from above is that the shadows created

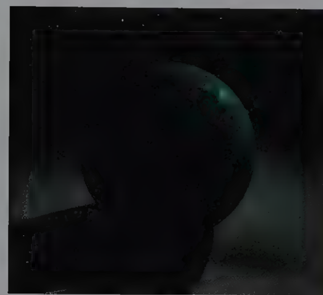

PAIR AT A 45-DEGREE ANGLE

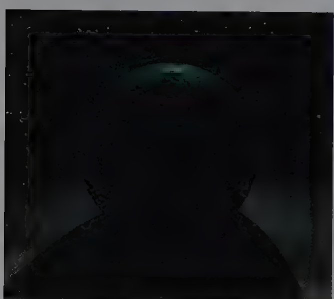

FRONTAL FROM BELOW


FRONTAL AT SUBJECT'S LEVEL

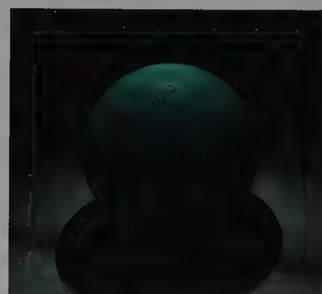

LATERAL AT SUBJECT'S LEVEL


LATERAL ABOVE SUBJECT


LATERAL ABOVE, BEHIND SUBJECT


BACK ABOVE SUBJECT


BACK SIDE OF SUBJECT


OVERHEAD

by the main subject are usually projected on the floor and not on other objects in the scene, as would be the case when using frontal lights from below or lateral lights at the subject's level.

A **lateral light from above and behind** is an effective way to outline a subject against a background. This position combines some of the advantages and limitations of all lateral, overhead, and back lights. It models the actor with contrasted shadows (lateral), and it also creates a halo of light on the top of the subject that clearly differentiates it from the background (back and overhead). Both the **side back light** and the **top back light** create halos or rims of lights on the subject's edges. Back lighting can be an effective way to create depth in a scene. **Overhead lights** also create dramatic halos around the top of a subject. As with back lights, overhead lights can also add depth and drama to a scene (Fig. 8.6.6).

8.5.3 A pair of lights at a 45-degree angle (one stronger than the other) delineate the edges of the delicately shaped legs of this insect and make its hair glow. (Courtesy of Akira Kai, FOTON.)

8.5.4 (Opposite page, top) The soft area lighting from below gives these dancing robots a light and sublime look. (*Aerobot*, © POLYGON PICTURES/IPA/NK-EXA.)

8.6 Getting Ready

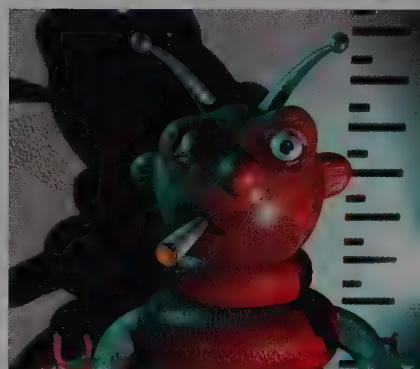
Learn About Traditional Lighting

Learning about traditional light sources can give you ideas on how to better use lighting in a computer-generated scene. Some of the main differences between lights used for motion picture, theater, and television production include the color of light emitted by the lamp, the power of the lamp, the number of lamps, the shape and surface of the source, and features like reflectors, lenses, and barndoors.

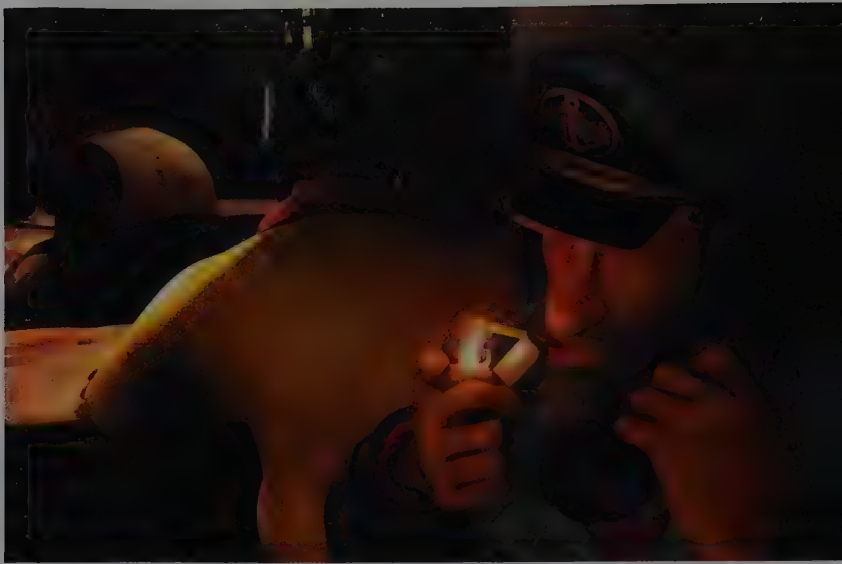
The power of the lamp is measured in watts but the amount of light emitted is measured in footcandles. Figure 8.6.1 lists the difference between light output by a similar lamp at different distances and in two different source configurations. Notice how the focused beam of a spotlight outputs many more footcandles than the more diffuse light of a floodlight source. Light sources of 10,000 watts (10K) or more are used for large sets and are usually fairly large in size and surface. Light sources between 2,000 and 5,000 watts are used for medium sets. Lights at 1,000 watts or less are versatile because they can be placed almost anywhere in the scene, and are often used to add lighting accents. Most light sources have one lamp—but a few have groups of lamps, usually in the form of a rectangular matrix, or rigs of fluorescent tubes. As far as the shape of light sources goes, most spotlights are round and floodlights are rectangular or round. **Reflectors**

consist of a highly reflective material inside of the light source that helps to amplify the amount of light emitted. Lenses and **barndoors** are used to focus and shape the beam of light. **Fresnel lenses** have concentric lens sections, and are especially effective at focusing sharp beams of light from light sources that have a large diameter. All kinds of diffusion materials and bounce cards are also used to direct light.

The traditional lighting sources, listed in Figure 8.6.3, are often referred to by the watt power of their lamp—1K stands for 1,000 watts. **Tungsten fresnels** are effective at creating a sharp and intense beam because of their built-in reflector and lens. The 2K fresnel has been widely used as a key light on a single actor. **HMIs** are also widely available with fresnel lenses. **Xenons** are very powerful lights with the highest ratio of lumens-per-watt of all light sources, and they emit a flicker-free light that makes them suitable for high-speed cinematography. Open face lamps, as their name indicates, have no lens to focus the light, but they do have a wide reflector that spreads a lot of light. **Open face lamps**, also known as skypans, are often used to bounce light off other surfaces, and their output can be controlled with barndoors. The PAR light sources are similar to car headlights because the lamp includes a self-contained bulb, lens, and reflector. **PARs** are often used in groups of five, nine, or twelve lights; the 9-light version, for example, is a matrix of 3×3 lamps. Some PARs are available with interchangeable lenses and with different light spread characteristics. **Softs** have different characteristics



8.5.5 The lateral lighting projects a harsh shadow on this smoking creature who wonders what his luck will be. (© Jim Ludtke.)



8.5.6 The lighter is an example of a practical light shining from the front and below. (© 2002 Avalon/ Sparkling*.)

Light Output in Footcandles of HMI Fresnel Sources

Spot	@ 10 ft.	@ 20 ft.	@ 30 ft.
12K	45,000	16,000	7,300
6K	49,000	12,250	4,400
4K	28,000	7,150	2,500
2.5	13,200	3,450	1,200
1.2	7,150	2,230	670
575	1,150	706	228

Flood	@ 10 ft.	@ 20 ft.	@ 30 ft.
12K	5,800	1,650	730
6K	4,100	1,000	370
4K	2,000	510	195
2.5	1,800	410	150
1.2	490	168	54
575	300	64	19

8.6.1 This list compares the illumination of a popular type of light for film and video, the HMI fresnel source, used both as a spotlight and a floodlight. The distance to the subject is measured in feet, and the illumination is measured in footcandles. Notice the large difference in intensity between the spotlight and the floodlight versions.

that diffuse their light, ranging from frosted lenses or lamps (like fluorescent tubes) to white (instead of silver) reflectors. **Broads** are rectangular reflectors often used as powerful fill lights or footlights on a theater stage. Last but not least, **lekos** are the highly focused theatrical spotlights often used to project patterns (called gobos) on the set.

Minimize Rendering Time

Try to minimize rendering time by keeping the number of lights down to a minimum. Most scenes can be properly lighted with a couple of well-placed light sources. Experienced lighters always try to create lighting impact in a scene with the smallest possible number of light sources. Rendering time can be saved by studying the scene and placing only those light sources that are essential to the effect sought. Large numbers of light sources should be reserved for shots that are critical, dramatically or visually speaking (Fig. 8.6.2).

Check the Default Light

Check whether the rendering software that you use automatically creates a default lighting setup. Many programs use an ambient light source as the default that lights all the objects in the scene uniformly. Default ambient lights can be modified when the original settings are not suitable to the scene in question.

Invisible Light Sources and Missing Shadows

Do not forget to make the computer-generated light sources (the lamps, not the light) invisible after they have been defined and posi-

8.6.2 This rendering of three-dimensional lettering recreates the shadow effects that are typical of colored theatrical lights. (*Dancing Love* © Toshifumi Kawahara/POLYGON PICTURES.)

tioned in three-dimensional space. Otherwise, they will show—usually in the form of small boxes, arrows, or brilliant dots—in the final rendered image. Sometimes an object that is meant to cast shadows does not do so even though the light source has been defined as a shadow-casting light. This problem might occur because the shadow-casting preference has been turned off either in the object itself or in the shading technique or in both.

Simulated Shadows

Transparent planes can be used to create shadows on objects in the scene, and even on photographic backgrounds that are composited, or blended, with a three-dimensional environment. In the latter case the transparent planes have to be aligned with elements in the background through a series of trial and error alignment tests. (For more information on image compositing read Chapter 13.)

Lighting Is Related to Shading and Compositing

Much of the final lighting effect in computer-simulated environments is determined by the shading technique or techniques that are used to render a scene. This is especially true in shots where the computer-generated lighting must match the live action lighting (Figs. 8.6.4 and 8.6.5). In this chapter we limit the discussion of lighting to the elements directly associated with the process of lighting: the light sources, their lighting characteristics, and their positions in three-dimensional space. Chapters 6, 9 and 14 show how to modify the effect of the original lighting setup through rendering and compositing techniques (Figs. 9.9.1, 9.9.2, 14.1.4, 14.2.7, and 14.5.1).

Traditional Lighting Sources

Tungsten Fresnels
200–250 watts (inkie), 500–750 (tweenie), 1K (baby or ace), 2K (deuce or junior), 5K, and 10K

HMIs
575 watts, 1.2, 2.5, 4K, 6K, 8K, 12K, and 18K

Xenons
75 watts, 1K, 2K, 4K, and 7K

Open Faced Units
600 watts, 650, 1K, 2K, and 5K

PARs
650 watts (PAR 36), 1K (PAR 64); 575, 2.5K and 4K (HMI PAR)

Softs
1K and 2K (zips); 2K, 4K and 8K (studio softs); cone lights; fluorescent rigs

Broads
750, and 1K

Lekos
1K, and 2K

8.6.3 Popular lighting sources used in film and video production.

8.6.4 The computer-generated lighting on the computer-generated pet was carefully matched to the soft lateral light above the young actor in *Spy Kids 2*. (© 2002 Hybride. Images courtesy of Dimension Films.)

8.6.5 The lighting on this character from *The Cathedral* accentuates the camera's depth of field, and brings up the texture detail. (© 2002 Tomek Baginski and Platige Image.)

8.6.6 The reflections of red neon light on the character were achieved by ray-tracing the textured plane. (*Teenage Mutant Ninja Turtles* and TMNT are trademarks and copyrights of Mirage Studios, Inc. *TMNT* © 2007 Imagi Production Ltd.)

Chapter 8

Key Terms

45-degree angle spotlight pair
Additive color model, system
Altitude
Ambient light
Area lights
Azimuth
Back light
Background
Barndoors
Beam angle
Bleeding
Bounce light
Brightness
Broads
Candela
Caustics
Color
Colored lights
Cone of light
Decay
Default lights
Diffuse inter-reflections
Directional lights
Exclusive link
Eyelight
Fall-off
Fill lights
Footcandle
Four-point lighting
Fresnel lenses
Frontal light at the subject's level
Frontal light from below
f/stop
Global light sources
Glow
Gobo lights
Halo
HMIs
Illumination level
Inclusive link
Infinite lights
Intensity
Invisible lights
Lekos
Key light
Kick light, kicker
Lateral light, at subject's level, from above, and behind
Lens flare
Level of illumination
Light measurements on location
Light shader
Light vector
Light-based colors
Lightning
Linear lights
Linked light sources
Local light sources
Lux
Main action area
Metercandle
Moving lights
Number of light sources
Numerically
Omnidirectional light
On location
Open face lamps
Ordinary lighting
Orientation
Overhead light
PARs
Penumbra
Pigment-based colors
Point light
Position
Practical lights
Procedurally
Reflectors
Rim light
Secondary action area
Selective light sources
Shadows
Side back light
Softness, Softs
Spherical coordinate system
Spotlight
Spread
Stage
Special lighting effects
Sound stage
Subtractive color system
Sun
Three-point lighting
Tinted lights
Top back light
Tungsten fresnels
Umbra
Visible light sources
Visually
Volumetric lights
Zone system
Zone V
Xenons

8.6.6 A single overhead light cascades onto the subject's hair and reveals the outline of her body while keeping the rest in the penumbra. (*Bondex, The Fashion Parade*. Director: Majid Loukil. Agency: Callegari-Berville. Images courtesy of Ex Machina.)

9.1.1 The human characters and some objects in *A Gentlemen's Duel* were rendered with ambient occlusion for increased detailed shading. A variety of hand-painted maps were used (color, bump, normal, specular, subsurface scatter, and wetness), and subsurface scattering was used for the skin shader available in SplutterFish's Brazil renderer. (Created by Blur Studio, Inc.)

CHAPTER 9

Shading and Surface Characteristics

QUICK INDEX

(Image courtesy of Andy Boyd.)

Summary

SHADING THREE-DIMENSIONAL SURFACES is done by calculating the effect of light on the objects in the scene. The focus of this chapter includes the main shading and mapping techniques, light reflectance models, and the characteristics of three-dimensional surfaces including reflectivity, color, texture, and transparency.

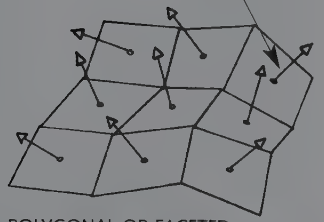
9.1 Surface Shading Techniques

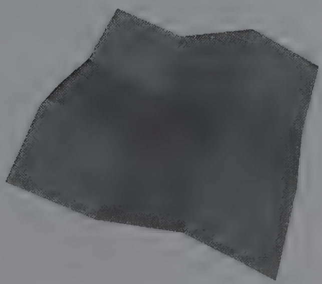
The visual appearance of objects and environments is determined through the shading process. The shading process creates surfaces on the polygonal geometry submitted to the rendering process created during the modeling process. (For the purpose of shading most software programs convert surfaces of different types to polygonal surfaces.) Surface shading is calculated based on the position of objects and distance relative from the light source, and it also takes into account the surface characteristics of the objects (Fig. 9.1.1).

Shading is the moment in the rendering process when visible surfaces are assigned a **shading value**. This value is calculated based on the relationship between the surface normals and the light sources that reach the surface. **Surface normals** are vectors, or straight lines with a specific direction, perpendicular to a polygonal surface. The surface normals are used to define the orientation of a surface, and they have a paramount role in the calculation of surface shading, as well as determining whether surfaces are visible or hidden. **Vertex normals** are located on the vertices, or corners, of polygons and they are also used in the shading process.

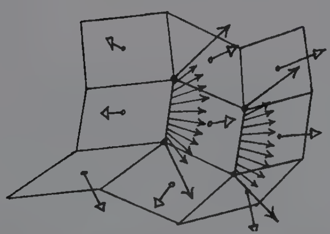
A Light Reflectance Model

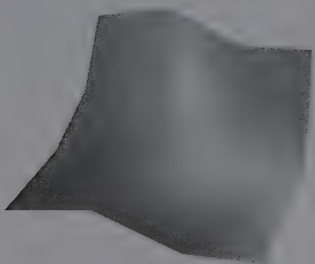
Shading techniques are based on representations of light and surface called **light reflectance models**. Some of the early shading techniques used clever shortcuts to calculate realistic-looking shading





9.1.2 The shading of each polygon with faceted techniques is determined by one surface normal per polygon.





9.1.3 In specular shading, the vertex normals are interpolated across the surface of the polygon, and then shading is calculated at each point on the surface.

without having to take into account all the elements that would be part of an accurate and consistent representation. Most software today uses a general light reflectance model that describes how light is reflected—from a surface under **local illumination** conditions. This mathematical model is called **BRDF**, short for **Bidirectional Reflectance Distribution Function**. Based on the amount of light reaching each point on a surface, the BRDF calculates how much light is reflected and in what direction (Fig. 9.1.5). Most general shading models based on BRDF calculate diffuse, specular, and mirror-like components. These components can be used to analytically approximate the reflective properties of materials and hence define material shaders. There are additional specialized shading models also based on BRDF used to describe unique surfaces—for example, with extreme roughness or with anisotropic characteristics. Three popular surface shading techniques using local illumination include diffuse, specular, and smooth shading. Shading techniques are sometimes also referred to by the last name of the individual who authored the technique.

Each rendering program offers its own version of the general shading model, and on occasion the shading models are mixed with each other or modified—for example, by adding new variables into the shading equation—resulting in **hybrid shading models**.

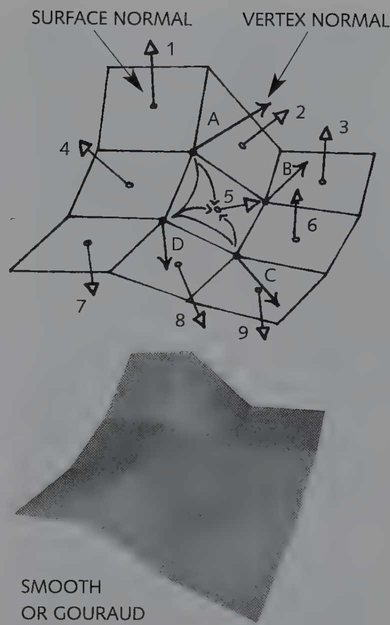
Diffuse Shading

This technique assigns a single and constant shading value to each visible polygon on the surface according to the angle of its normal in relation to the light source. **Diffuse surface shading** usually assigns a constant shading value to each polygon on the surface, resulting in a **faceted appearance**. For this reason, diffuse shading techniques are sometimes called polygonal shading, or constant value faceted shading (Fig. 9.1.2). Diffuse shading measures the amount of light received at the center of the polygon, but some variants measure the light received at the vertices of each polygon and apply the average value uniformly to the entire polygon. Diffuse shading techniques are insufficient for representing complex surface characteristics such as texture and transparency. Diffuse shading is the simplest type of shading and also the fastest because it uses only one surface normal per polygon to determine the shading for the entire polygon. The **Lambert shading model** is a popular form of diffuse surface shading.

Specular Shading

Specular surface shading techniques create surfaces with highlights typical of reflective surfaces. The word *specular* means mirror-like. In addition, specular shading techniques create a smooth continuous shading across polygons by using normal interpolation techniques. Specular shading is also called normal vector interpolation

shading because it calculates the shading at every point on the surface of a polygon. This is done by interpolating the vertex normals and shading every point on the surface of the polygon by computing the relation between the angle of its normal and the angle of the incident light (Fig. 9.1.3). The **Phong shading model** is the original version of specular shading developed in 1975, with additional variations by Blinn and Cook. This technique takes into account ambient, diffuse, and specular shading, which can deal with detailed surface characteristics. This technique creates more accurate renderings than the diffuse or smooth techniques, but it is also more computationally intensive.



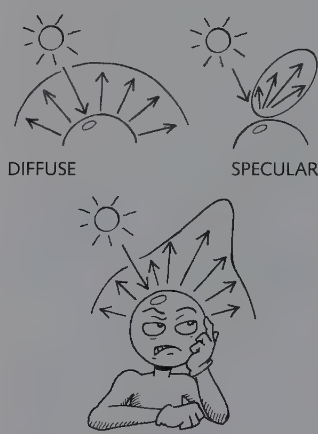
9.1.4 In smooth shading, the surface normals of four adjacent polygons are averaged to determine the four vertex normals of a single polygon. This process occurs four times in this illustration. Surface normals 1, 2, 4, and 5 are averaged to create vertex normal A, and so on. Finally, the intensity values of the vertex normals are interpolated across the polygon.

Smooth Shading

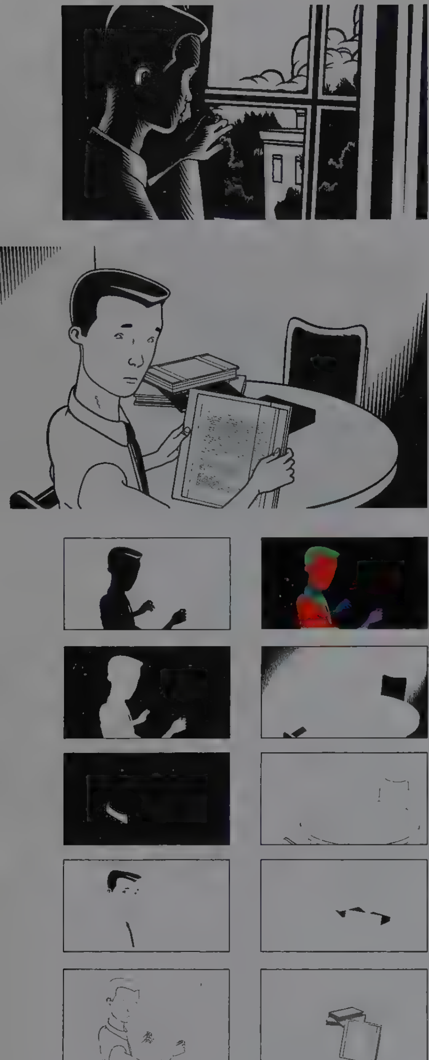
This efficient technique uses a clever method to assign a continuous shading value that blends across the polygons on a surface. The basic idea behind this technique is to average the surface normals of adjacent polygons, creating the appearance of a smooth transition of shading between polygons, even those that have a small amount of modeling detail. **Smooth surface shading** first samples the amount of light reaching the surface normals in the center of polygons, then creates a vertex normal from the averaged values of the surface normals of adjacent polygons, and finally blends the intensities of the vertex normals in a polygon (Fig. 9.1.4). Smooth shading is also called intensity interpolation shading or the **Gouraud shading model**. Smooth shading takes into account the ambient and diffuse parameters but does not compute the highlight values that are typical of reflective surfaces. It is an efficient shading shortcut but not sufficient to yield realistic and consistent renderings.

Ambient Occlusion Shading

Ambient occlusion recreates the lighting effect of a luminous but overcast day, including the gradual soft shading of objects combined with well-defined deep shadows (Figs. 8.1.10, 8.2.4, 9.1.1, 9.4.5, and 12.2.12). **Ambient occlusion** shading calculates the amount of light not reaching a surface because it is blocked, or occluded, by other objects in the scene. Ambient occlusion is generally calculated by casting omnidirectional rays from the surfaces being shaded toward a virtual sky, usually a dome surrounding the scene. The brightness of the illumination on the surface is defined by the rays that reach the sky unobstructed: the rays that are obstructed by any geometry are not taken into account into the illumination calculation. Areas on the shaded surface that receive no direct sky illumination are rendered dark, and properly rendered ambient occlusion scenes have light and dark values spread throughout their histogram. Ambient occlusion is reminiscent of global illumination but is simpler and efficient, it can be considered a hybrid shading model because it combines components of both local and global illumination shading.



9.1.5 The BRDF model includes diffuse and specular components.



9.2.1 The non-photorealistic shader used in *Fear(s) of the Dark* replicates the high contrast shading style of illustrator Charles Burns, with line patterns defining the morphology of surfaces. An earlier approach (top) offered automatic orientation of the lines and worked well on simple objects but was rigid to work with. The final shader offers the ability to fine tune shading and graphic details, and relies on multiple passes. (© Prima Linea Productions, www.FearsOfTheDark-themovie.com.)

RenderMan Shading

RenderMan is a collection of rendering tools that includes a shading language and a renderer or rendering program. The shading language is a C-like programming language. It can be used to describe how things look, and it is especially well suited to create new looks and complex appearance with simple geometry. One of the main innovations of the **RenderMan shading language** is the fact that users have the ability to extend the **renderer** by writing shaders and shading capabilities with its shading language. As seen later in this chapter, a shader is a collection of shading characteristics and rendering techniques that are applied to an object during the rendering process (Fig. 9.4.2). There are four basic types of RenderMan shaders: surface shaders (probably the most widely used type), light shaders, volume shaders, and displacement shaders. Shaders can be applied in layers as illustrated in Figure 9.2.4.

RenderMan can handle most types of geometry including polyhedra, parametric curved surfaces, patches, NURBS, point clouds, blobby implicit surfaces, and subdivided surfaces that have arbitrary topology, smooth interpolation, controllable creasing, improved motion blur, particles that have the ability to reflect light toward the camera, and soft shadows that fade out in the distance. RenderMan shaders written for general use can be saved as templates or library subroutines and incorporated as modules in other projects. Special-purpose shaders are oftentimes customized for a particular geometry or production pipeline and therefore are not suitable for recycling (Figs. 1.2.7. and 1.3.2). In addition to the shading, or scene description language, Pixar released a rendering software called **Photorealistic RenderMan,** also known as **PRMan**. This software has popularized the use of **RIB** files, short for **Renderman Interface Bytestream**. RIB is a renderer-independent scene description file format using the RenderMan language. Many turnkey and Open Source renderers today are RenderMan-compliant and use the RIB file format to exchange scene and frame information.

9.2 Surface Shaders and Multi-Pass Rendering

A few rendering programs provide a convenient way to render all or part of a three-dimensional scene by grouping in a single shader multiple variables that influence the shading. A **surface shader** is a collection of the surface characteristics and shading techniques that are applied to an object during the rendering process. Surface shaders are used to define the **surface finish** of the **simulated material** that a three-dimensional object is made of. The basic surface characteristics contained in most shaders include reflectance of light, color, texture, and transparency.

Surface shaders are used to determine the amount and color of light that is reflected by three-dimensional surfaces. Shaders also represent a flexible way to manage the large number of variables

that are used to render three-dimensional objects. It would be ideal if shaders were defined in a portable way so that different rendering software could interpret them in a consistent way. But in practice there are a fair number of shader proprietary formats.

The number of rendering methods available in different software varies between programs—some offer several rendering methods, while others offer just a few. As explained earlier, each rendering software offers a unique collection of techniques and methods to render three-dimensional objects. The concept of a shader has slightly different meanings in different programs. In general, it means a collection of surface characteristics and shading techniques defined in a consistent way. Sometimes rendering programs provide all the information that is usually contained in a shader but do not call it a shader; instead, one has to find the multiple shading parameters scattered throughout several dialog boxes. This latter approach may be less convenient than working with shaders that consolidate all variables in a single place and offer easy editing.

Sometimes shaders include only surface characteristics that can be applied to the three-dimensional objects independently from the shading techniques used to render them (Fig. 9.2.1). When shaders include only the characteristics of the surface material and not the shading technique to be used for rendering they are often called **surface libraries** or **material databases**. Shaders are usually applied to entire objects or groups of objects, but they can also be assigned to parts of an object—for example, a group of polygons as shown in Figure 12.1.7. In general, surface shaders use some or all of the following information to determine the shading of a surface: shader name, shading technique and parameters, surface characteristics and parameters, and rendering method and parameters (Fig. 9.2.2). Shaders can be edited by typing numerical values, by dragging sliders, or by modifying function curves. The ranges of shader values are often between 0 percent and 100 percent, from 0 to 255, or from 0 to 1.

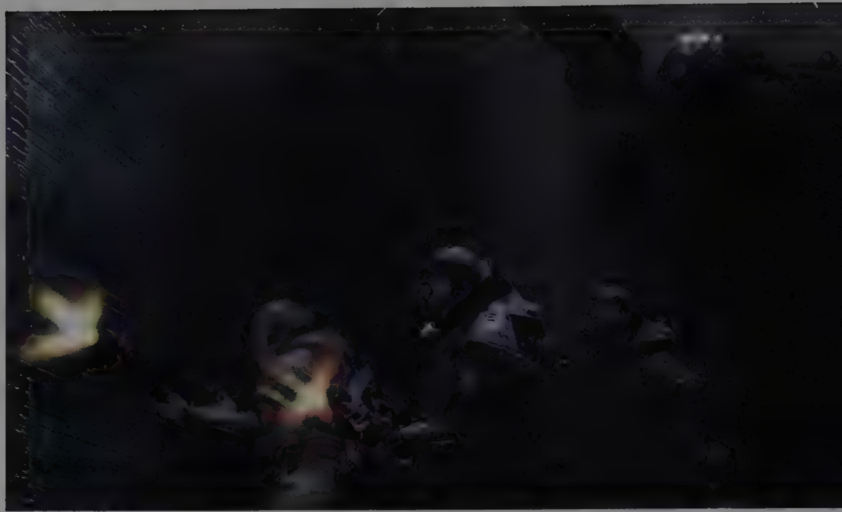
9.2.2 This image of brushed metal was created with a reflectivity model for metallic surfaces. The model handles the effects of diffraction by taking into account the wave nature of light, and it was implemented in the form of Maya software shaders. The directions of anisotropy—different values reflected in different directions—were texture-mapped to add visual detail. (© 1998 Jos Stam, Alias|Wavefront.)

Specifying surface characteristics often requires a fair amount of attention to detail. Doing a good job in the simulation of surface materials has an impact on the quality, refinement, and energy of the final rendered image. Specifying the surface characteristics and choosing the shading techniques are two distinct steps, but they are intrinsically related to one another and often overlap with each other.

When defining surface shaders and when applying them to three-dimensional surfaces, it is important to consider the lighting characteristics that the shaders will be used under. Lighting conditions have a powerful effect on the appearance of shaders, to the extent that the same shader can look very different under two different lighting conditions.

Rendering in Layers and in Multiple Passes

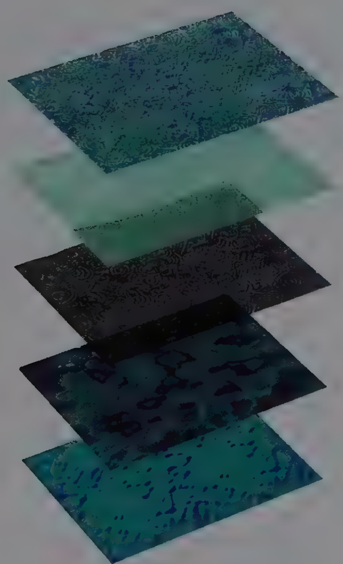
One of the great conveniences of today's rendering techniques is the possibility of rendering surface shaders each in their separate **render-**



9.2.3 Still from the movie *Final Fantasy: The Spirits Within* rendered with multiple layers of two-dimensional maps and three-dimensional lights and geometry. Motion capture was used for about 90 percent of the human motion, using a 16-camera Motion Analysis system in an 8 x 10 x 2.5 meter capture area, resulting in over 2,000 motion capture elements. Only body data was captured, not including hands and faces. (© 2001 FFFP.)

ing layer. Rendering a three-dimensional scene in multiple separate layers, or **rendering passes**, provides flexibility in determining the final look of the surfaces. Keeping each set of surface characteristics in separate layers facilitates changing just one or two layers while keeping the others intact. Commonly used rendering passes, layers, or maps include Z-depth, shadows, diffuse, specular, ambient occlusion, color, displacement, and transparency (Fig. 9.3.7).

A single surface can also be separated in multiple surface layers. The method of surface layering is reminiscent of the way in which painters during the Renaissance created their paintings in layers, for example, by first priming the surface with a white mixture, and then applying, one at a time, multiple coats of opaque and transparent paint and varnish. On occasions the **underpaint** coats contained colors that were not directly visible but that influenced the translucent colors from upper layers in particular and the overall color effect in general. At the end of this process the dry surface was burnished to compact all the layers of paint and varnish, and also to create a smooth, shiny outer surface. There is a great variety of techniques for layering surface shaders, just as there are many techniques for painting (Figs. 9.2.3 and 9.2.4). **Surface layers** can also be used to simulate a simplistic version of the subsurface scattering technique described later in this chapter.



9.2.4 The surface in the top layer is composed of the four separate layers below it. The second layer from top to bottom is translucent and refracts the image of the layers below (see the top edge of the third layer from top to bottom). The different layers of a planet's atmosphere or a human's skin can be easily rendered using a layering technique.

9.3 Image Mapping

Image mapping is an important component of the surface shading process, and it is also a rich technique that deserves to be examined separately. The basic idea behind **image mapping** consists of taking a two-dimensional image and mapping it onto the surface of a three-dimensional object. There are many mapping techniques—for example, projecting or wrapping—and each creates a distinct result. But

9.2.5 These detailed textures in the interior of a Lamborghini Reventon are rendered in real time with an Nvidia graphics card. (© RTT AG.)

the real power of image mapping lies in the fact that two-dimensional images can be used to efficiently simulate not only the texture of a three-dimensional surface, but also other surface attributes such as reflectivity, and roughness.

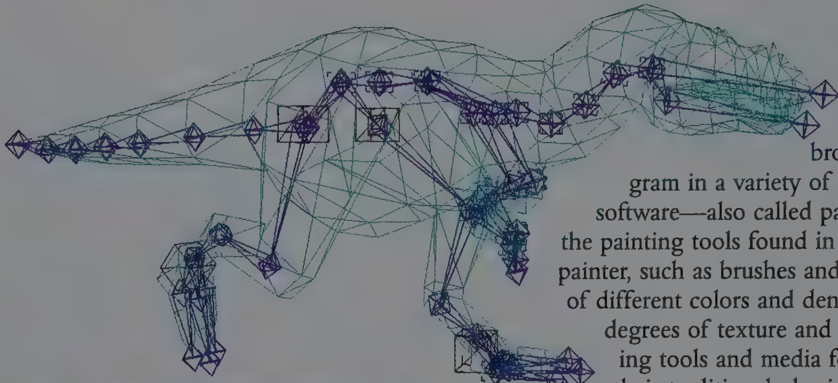
Image mapping techniques are often used as shortcuts for simulating surface characteristics. In fact, the exact same image can be mapped in different ways onto a surface for simulating attributes such as color, texture, and transparency (Fig. 9.3.1). Each of these attributes can also be simulated with a unique image map.

Image maps can modulate surface characteristics by linking the brightness or color of a pixel in the image map to the characteristics of the point in the surface where that pixel is mapped. For example, the brightness of a pixel in an image map can control the reflectivity of the point on the surface where the pixel is mapped, or its color, or its transparency. Different image maps can also be combined to control different aspects of the surface characteristics of an object. The types of image maps covered in this chapter include reflection and environmental maps, color maps, procedural maps, bump and displacement maps, and transparency maps. The nomenclature used here to describe image maps is quite generic and is used in many three-dimensional software programs.

9.3.1 The same two-dimensional image is used as a map to control different surface attributes including color, two types of texture (bump and displacement), and transparency.

Creating the Map

Two-dimensional images that can be mapped onto three-dimensional surfaces include painted images, photographic images, and abstract patterns. As explained shortly, each of these images is best suited for



a specific purpose. Image maps can be created directly with computer paint systems and brought into the rendering program in a variety of file formats. **Digital painting** software—also called paint systems—provides many of the painting tools found in the studio of a traditional painter, such as brushes and sticks of different types, paints of different colors and densities, and papers with different degrees of texture and absorbency. These digital painting tools and media for the most part behave like their traditional physical counterparts, especially when a **pressure-sensitive** graphics tablet is used instead of a mouse. Painted images to be used as maps can be created from scratch with the computer paint system, or they can be based on a sketch or photograph that is scanned into the program (Figs. 9.3.2 and 9.3.4). Images for mapping can also be captured directly into the computer system with a variety of input devices. This includes recording a live image with a **digital camera**, scanning an existing photograph or painting with a flatbed or laser **digital scanner**, or generating an abstract pattern with procedural software (Fig. 9.6.4). Scanners and digital cameras transform the visual information into numerical information that can be easily processed by the computer software. This conversion is done by converting the **continuous visual information** that we find in either reality, a color photograph, or a painting into a series of **discrete numerical values**. This conversion is based on an **averaging** of the values found in the original image. The scanning process starts with a **sampling** of the color values in the image. The number of samples taken from an image directly determines the spatial resolution of the image map.

9.3.2 Different color maps were painted for different sections of a T-Rex low-resolution polygonal model. The game engine takes care of mapping the different sections of the image onto the right area of the geometry. The inverse kinematics skeleton of a real-time character are shown over the wireframe skin. Notice the large amount of joints on the tail and neck, and the importance of the hips and feet in effecting motion. (Courtesy of Angel Studios.)

Maps for Real Time

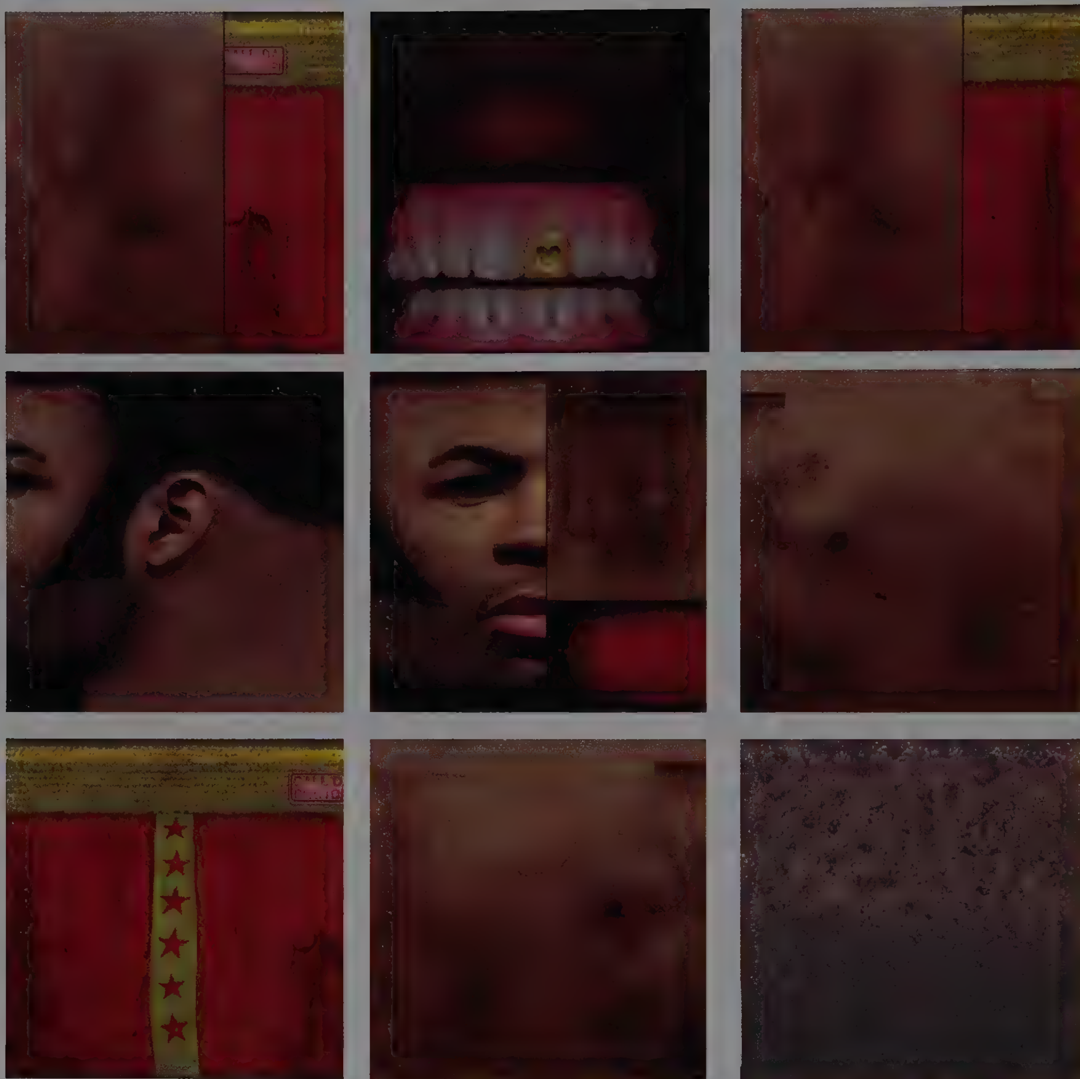
When creating maps for real-time applications, the main concerns are making sure that they look good, and also that they are efficient and compact enough to be rendered without hindering the performance of the game, online, or virtual reality experience. There is no set limit to how large maps for real time should be; each playback system and each project have unique limitations. Usually maps for real time are developed as series of small panels or tiles; 256×256 or 512 x 512 pixels are common sizes. Each of the panels may contain different images that are usually assembled and mapped onto the model in real time. Sometimes a single panel will be repeated—or tiled—to fill a large surface. The character in Figure 9.3.4, for example, uses almost a dozen 256×256 panels. Some of the panels, the hair panel for example, are tiled over large surfaces. A basic principle when tiling image maps is to make sure that the texture seams or outer borders match, so that the resulting texture will appear to be a large single map as opposed to many small off-sync tiles.

9.3.3 This rendering of a virtual movie actress from the 1930s (can you guess who?) was created by mapping different types of texture maps onto spline-based geometry. The color map used can be seen in Figure 9.5.1, the bump map in Figure 9.6.1, and the geometry in Figure 5.1.3. It was animated with motion capture using 80 facial markers. (Director: Daniel Robichaud, Animation Supervisor: Stéphane Couture, Art Direction: Michelle Deniaud. © 1999 Virtual Celebrity/ Marlene Inc.)

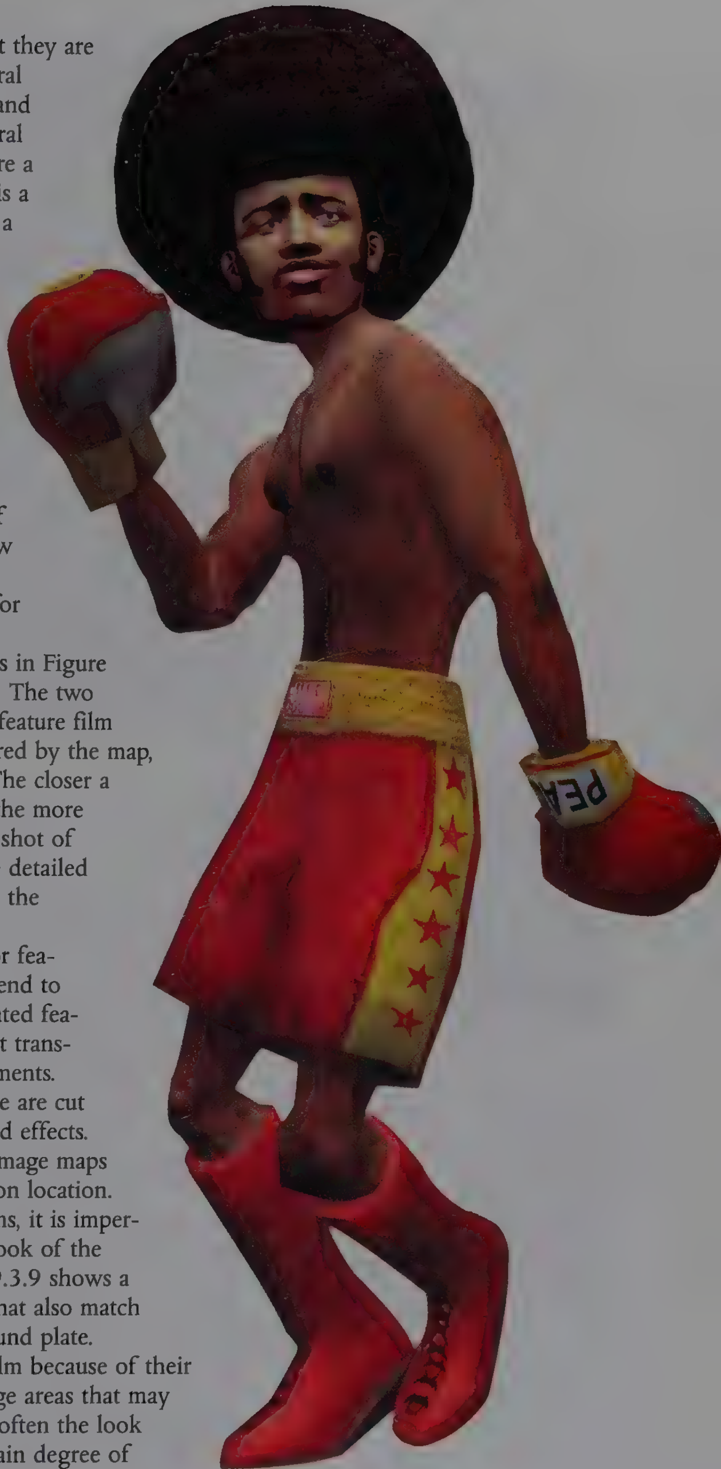
The maps used in the real-time character shown in Figure 11.1.1 have a surface of only 32 × 32 pixels, which translates into a total file size of 3.7 Kb. But after conversion to the Playstation .sif texture format, the bitmaps shrank down to 572 bytes. The main character in the *Spyro the Dragon*™ game uses about 20 of these texture maps, while most of the supporting cast uses up to 10 maps each. In fact, the majority of the characters in this and similar games have texture maps that cover a portion of their body surfaces, while the rest of their bodies are rendered with smooth shading only, in order to conserve precious memory—VRAM in particular—during gameplay.

Generally speaking, the painting style used to develop image maps for real time should be kept simple, and usually within a limited color palette—especially if color lookup tables will be used in the rendering of the game (Fig. 9.3.5). As seen in Figure 14.1.5 the use of color lookup tables for rendering tends to limit the range of color, even if the original image map had rich and subtle color values. On the other hand, increasing the amount of detail painted in the texture map usually softens the look of low-resolution geometry. Deciding what the right balance is between too much detail and too little detail is often based on previous experience, on the visual style that was specified for the specific project, and on an understanding of what works best in the target playback platform.

Image maps for real-time applications are rarely layered due to



9.3.4 These panels of 256 x 256 pixels contain the texture maps for the character rendered on the opposite page. (READY 2 RUMBLE™ BOXING © 1999 Midway Home Entertainment Inc. All rights reserved. Likeness of Michael Buffer and the READY TO RUMBLE® trademark used under license from Buffer Partnership. All character names are trademarks of Midway Home Entertainment Inc. Midway is a trademark of Midway Games Inc. Used by permission.)



the limitations of the real-time rendering, but they are oftentimes used in conjunction with procedural textures of the kind shown in Figures 9.6.4 and 9.6.5. In those cases when real-time procedural textures are not feasible—because they require a fair amount of computing—*baking* a texture is a common shortcut that consists of generating a procedural texture, saving it as an image map, and applying it onto the geometry.

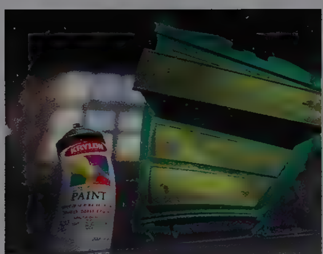
Maps for Feature Film

When creating maps for feature film the main and overriding concern is that they look good even when projected on a large theater screen; rendering times usually are of secondary importance. The size of image maps for feature film ranges from a few megabytes to dozens, even hundreds, of megabytes. The image map in Figure 9.5.1, for example, is 3072 × 3072 pixels and over 36 megabytes and was used along with the maps in Figure 9.6.1 to render the headshot in Figure 9.3.3. The two key factors in deciding how large a map for feature film should be are the size of the area to be covered by the map, and the distance of the map to the camera. The closer a map is to the camera, the larger its size and the more detail will be required. A map for a close-up shot of the main character is going to be much more detailed than a facial map for a virtual extra sitting in the background.

The style used to develop image maps for feature film is usually very detailed since these tend to be rendered accurately. Image maps for animated features often use a lush painterly style since that translates well into the three-dimensional environments. Sometimes portions of photographic reference are cut and pasted into the digital paintings for added effects. In the case of live action feature films, most image maps usually come from photographs, often taken on location. In most visual effects shots for live action films, it is imperative that the image maps closely match the look of the background plates and live elements. Figure 9.3.9 shows a creature rendered with detailed image maps that also match the coloring and lighting of the live background plate.

Image maps are rarely tiled for feature film because of their tendency to reveal a uniform pattern over large areas that may look unnatural. A trick that is often used to soften the look of tiled image maps consists of adding a certain degree of

9.3.5 Finished vehicle from *Hydro Thunder* and four of the six 256 × 256 image maps used on the surface. (© 1999 Midway Home Entertainment Inc. Midway is a trademark of Midway Games Inc. Used by permission.)



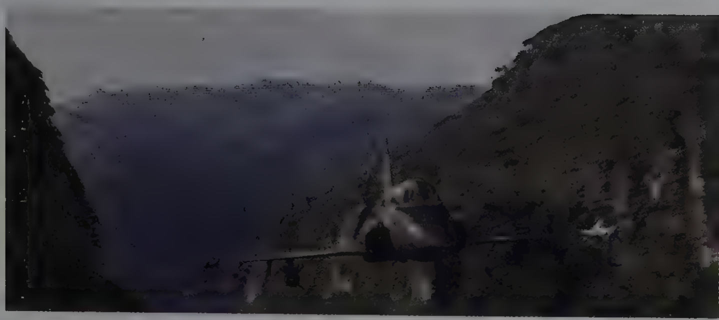
9.3.7 An image made of several composited layers, and two of the layers (opposite page bottom). The Z-depth layer in grayscale is used to create a depth of field effect on the layer with the window elements. (Images courtesy of Wyse Advertising and Will Vinton Studios. Sal DeMarco and Michael Chaney, Producers. © 1999 Krylon Products Group.)

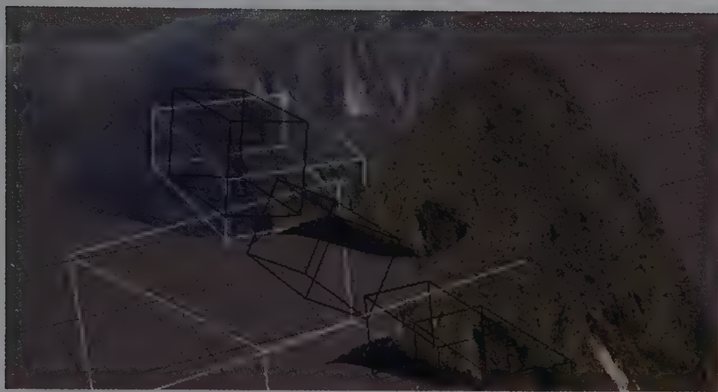
random image noise to the regular pattern or patterns that are being tiled. This technique was used effectively for rendering portions of the road's asphalt in the feature film *Toy Story 2*. Image maps for feature film are often layered to achieve a richer look, and they are also often used in conjunction with procedural textures. The latter provides an excellent solution to mapping convoluted geometry that might make the wrapping of the image maps quite difficult. Finally, the techniques of image-based rendering and modeling described in earlier chapters take the concept of photographic image maps to a new level that actually derives most illumination and modeling information from the photographs and measurements taken on location (Figs. 5.5.12, 6.8.1, and 6.8.2).

Projection Methods

There are many ways to project image maps onto three-dimensional surfaces. Some projection methods are simple and others are complex; some create realistic effects and others create surprising results (Fig. 9.3.6). Choosing a **mapping projection method** should be based on creative considerations without ignoring production concerns. Some projection methods may express the ideas behind a rendered image better than others. Some of the most useful projection methods include flat and cubical, cylindrical, and spherical.

The **flat projection** method applies maps onto surfaces in a flat



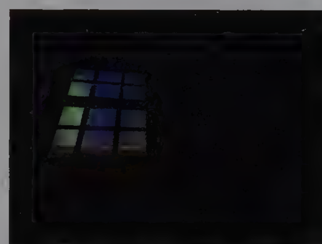


9.3.6 For this shot in the *Coronado* movie photographic images of the Santa Monica Mountains in Los Angeles were projection-mapped onto a simple terrain. Photographic images of a real F-5 fighter jet were textured on the computer-generated jet model. (*Coronado* images courtesy of Uncharted Territory, LLC.)

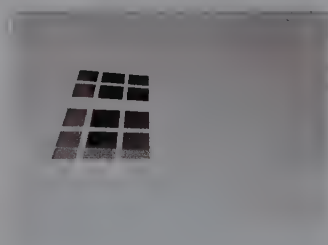
way. This projection method is ideal for applying image maps onto flat surfaces because the results are totally predictable, and the potential for distortion is minimal as long as the three-dimensional surface is parallel to the projection plane. In principle, flat projection can occur on any plane—XY, XZ, and YZ—with identical results as long as the three-dimensional surface is parallel to the projection plane. But flat projection can also be used on curved objects to simulate the effect of slide or film projectors because this method projects a flat image in a perpendicular way onto whatever is in front of it (Fig. 9.3.8). Another useful application of flat projection is the creation of **backdrops** and simple **dioramas** that include three-dimensional objects and characters placed in front of a painted or photographic backdrop.

The **cubical projection** method is a variation of the flat projection method that repeats the map on each of the six sides of a cube. This projection method is particularly effective with cubes but only as long as one of the planes of the cube is parallel to the projection plane. Cubical projection can also be used on curved or irregular objects to achieve unexpected results (Fig. 9.3.8).

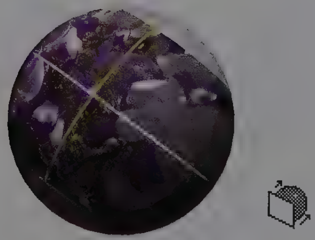
The **cylindrical projection** method applies maps onto surfaces by wrapping the sides of the map around the shape until the two ends of the map meet behind the object (Fig. 9.3.8). This projection technique is useful for mapping textures around elongated objects like a carrot or a glass bottle. Cylindrical projections are designed to wrap around the object and to cover its entire surface, but wrapping



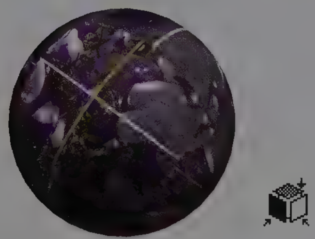
WINDOW ELEMENTS ONLY



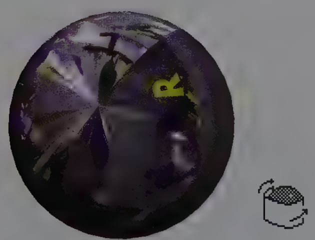
DEPTH MASK



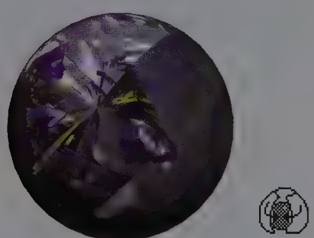
FLAT OR STRAIGHT PROJECTION



CUBICAL PROJECTION



CYLINDRICAL PROJECTION



SPHERICAL PROJECTION

9.3.8 An image map (previous page, lower left) applied to a sphere with four different projection methods. Notice how cubical projection repeats the map six times, and how spherical projection stretches the map toward the poles as it wraps it around. (Infini-D 3.0 icons. © Specular International, Ltd.)

of cylindrical projections can be customized so that the opposite sides of the map do not meet, and cover only a portion of the object. This is controlled by specifying the **angle of mapping** around the object with a degree value. A cylindrical projection can also be customized by determining whether the top and bottom of the object are to be left uncovered or whether they are to be covered with a cap. A cap in a cylindrical projection uses the same texture wrapped around the object unless specified otherwise.

The **spherical projection** applies a rectangular map by wrapping it around a surface until the opposite sides meet, and then pinching it at the top and bottom and stretching it until the entire object is covered. This technique is useful for projecting maps onto round objects, such as a basketball or a football. Spherical projections wrap the map around the entire three-dimensional object unless the projection is customized to cover just a portion of the object. This can be controlled—as it is in cylindrical projections—by specifying an angle of mapping (Fig. 9.3.8).

The **wrapping** projection method—as it is called in some programs—allows textures to be projected onto three-dimensional objects in a straight way, but also to be stretched until the four sides of the map are pressed against each other. This projection type is useful for placing texture maps over objects that may require stretching throughout the map for a good fit—such as terrains or complex surfaces. This technique is also effective for applying textures to small portions of three-dimensional objects, the same way that decals are applied to model airplanes (Fig. 9.3.10).

Positioning the Map

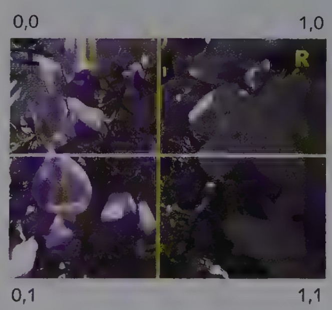
There are a variety of techniques that facilitate the placement of texture maps on three-dimensional surfaces. Ideally, maps should cover the entire three-dimensional surface unless a specific project requires a different approach. Texture maps are always **rectangular images** that are applied to polygonal or curved surfaces, and they can be defined by tagging their four corners. The nomenclature for identifying the corners of a texture map is simple whether the surface is made of polygons or curved patches. The upper left corner of the map is designated as the origin (0, 0), the lower left corner is (0, 1), the upper right corner is (1, 0), and the lower right corner is (1, 1). In some cases, the lower left corner is designated as the origin, and the upper right corner is designated (1, 1). Ordinarily a texture map is pinned by default to the origin of a surface, wherever the origin may be (origins can be located, for example, in one corner of the surface or in its center). This procedure is straightforward in cases when the three-dimensional surfaces are simple, and when the maps are supposed to cover the entire shape. However, placing texture maps on three-dimensional surfaces requires some fine-tuning when the surfaces are complex, when the proportions of the map and the surface differ, or when special effects are sought. It is important to

keep in mind that the tools for positioning the texture maps over three-dimensional surfaces vary greatly between programs.

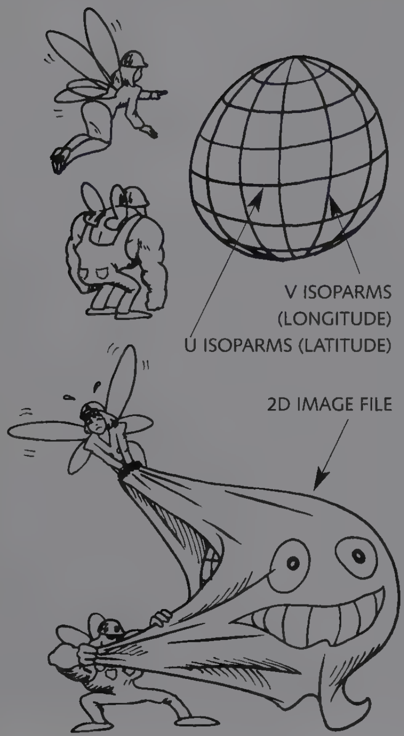
Some programs allow interactive placement of texture maps over three-dimensional surfaces. Texture maps can also be positioned very accurately over three-dimensional surfaces by inputting **numerical values**. With this method a map can be moved with precision over the surface. In the cases of polygonal objects, **XY coordinates** are used to **offset the map** over the surface. The default position of maps aligns them at (0, 0). If the map is offset by (.5, .5), it will be moved horizontally and vertically halfway across the surface.

In general, the **parameter space** used to position image maps on curved patches is based on the rectangular coordinate system used when maps are applied to polygonal models. But points on curved surfaces are defined in terms of their **UV coordinates** instead of their XY values. The parameter space of a curved surface is defined by a U (or u) horizontal value that stretches from 0 to 1, and a V (or v) vertical value that also ranges from 0 to 1. The value of U is commonly 0 on the left edge of the parameter space, and 1 on the right edge. The value of V is 0 at the top of the parameter space, and 1 at the bottom. All the points located within this rectangular parameter space are defined in terms of U,V coordinates. The rectangle is twisted and bent to match the shape of curved patches or quadric surfaces, which are built from a curve that is swept in three-dimensional space around an axis, like a surface of revolution. In quadric surfaces, the U axis represents longitude, and it runs

9.3.9 A three-dimensional computer-generated creature was composited with the live-action plate in this sequence of *Deep Rising*. Notice the matching of the computer-generated lighting with the lighting on the live set. (Image courtesy of The Walt Disney Company. © Hollywood Pictures Company. All rights reserved.)



MAPPING COORDINATES ON IMAGE



9.3.10 Wrapping projection is useful for fitting image maps with maximum coverage onto three-dimensional objects.

9.3.11 Characters for *Loco Rodeo*, created with organic shapes and image maps painted with traditional watercolor. (Image created by Mondo Media, San Francisco, CA. © 1997 The Locomotion Channel.)

approximately the circumference of the revolution; the V axis represents latitude, and it runs along the curve that is used to define the surface (Figs. 9.3.10, 9.3.11, and 9.3.12).

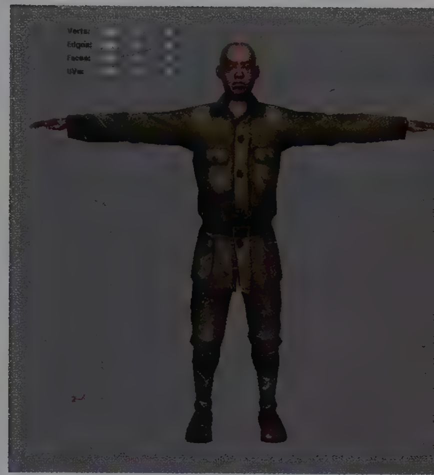
Positioning image maps with UV coordinates is a precise method that allows you to match specific pixels on the mapped image with specific vertices on three-dimensional curved surfaces and polygonal structures. Image textures that are mapped with UV techniques usually stretch like elastic surfaces—a silk stocking, for example—in a way that follows with little distortion the shape of the three-dimensional object. This is due to the extreme control of image pixel-to-object vertex matching offered by UV mapping.

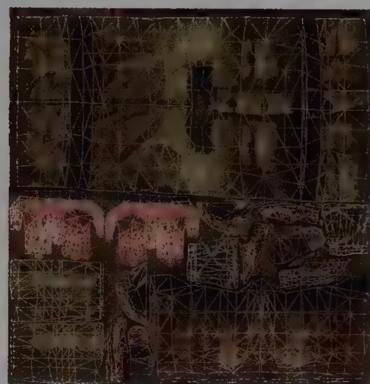
There are also techniques for controlling the map once it has been placed on the surface. These techniques include scaling and tiling. **Scaling** image maps can be used when the maps need to cover more or less of the surface of an object. **Tiling** an image allows you to create patterns based on repeating a tile or single rectangular image map (Fig. 9.3.15). A large number of three-dimensional software programs can repeat an image in a variety of arrangements along the vertical and horizontal axes. Some tiling permutations commonly available are plain repetition without any image flipping, repetition with horizontal flipping on every other tile only, repetition with vertical flipping only, and repetition with both horizontal and vertical flipping.

Map Blending

Map blending techniques determine the way in which surface layers, including image maps, blend with the surface of the object as well as with other surface layers. The blending of an image map with other surface characteristics can be controlled in a variety of ways. Some map blending techniques include overall blending, blending by types of illumination, blending with the alpha channel, and blending with matting techniques. **Overall blending** allows you to control the degree by which the image map blends uniformly with all the attributes of the surface. Overall blending is usually expressed in percentages of visibility, ranging from blending where only the map is visible to blending where only the surface is visible (and the map is totally invisible). Intermediate stages of overall blending allow for different degrees of blending. Blending by type of **surface illumination** controls the degree by which the map blends with the surface by splitting it in terms of ambient, diffuse, and specular areas of surface illumination.

Blending with **matting techniques** allows you to control the degree of blending using different parts or aspects of the image map as a mask. A **mask** is an image that masks or protects a surface, or portions of it, and it determines the degree by which different portions of the image map blend with the surface. Masks can be high-contrast or continuous. **High-contrast masks** have sharp edges and solid areas, while **continuous masks** have soft edges and different





9.3.12 An image map with the clothing for one of the *Medal of Honor* characters is placed on the polygonal surface using specific UV locations. (© 1999 Electronic Arts Inc. Medal of Honor and Electronic Arts are trademarks or registered trademarks of Electronic Arts Inc. in the U.S. and/or other countries. *Medal of Honor* is a trademark or registered trademark of Electronic Arts Inc. in the U.S. and/or other countries for computer and video game products. All rights reserved.)

shades of gray. Blending with matting techniques can be done by using all the pixels in the image map to mask out the surface, or by using only the black or white pixels in the map as a mask. Overall blending controls can be used in conjunction with matting techniques for creating a wide variety of blending possibilities (Fig. 9.3.13).

Blending or compositing with an alpha channel allows you to control the blending of the surface and the image map according to an additional image file that is used as a mask in the matting process. An **alpha channel** is a black-and-white image file that is linked to an image map. Alpha channels can be saved with an image file in the form of a fourth channel in a standard RGB image file. An alpha channel can be used to determine the degree of blending of the image map with a surface based on the brightness values of the pixels in the alpha channel. Total blending (transparency) can be assigned to the black pixels in the alpha channel, and lack of blending (opacity) to the white pixels; the reverse is also possible. Pixels with gray values are assigned different degrees of blending. (Read Chapter 13 for more information on matting techniques.) Map blending is often used in two-and-three-dimensional integration (Figs. 9.3.14 and 11.6.3).

9.4 Surface Reflectivity

Surfaces reflect light in different ways depending on a number of factors captured in the BRDF method described earlier. Surface

9.3.13 Blending with matting techniques can be used to reveal or cover portions of the image layers on the surface of an object. In this example the final result (top) was obtained by layering an image map and its own alpha channel onto a blue sphere, and revealing the dark blue surface of the sphere only through the black pixels in the alpha channel. A similar result can be obtained with a simpler image map without an alpha channel and using the white pixels in the map to reveal the dark blue surface underneath.

reflectivity, or reflectance as it is sometimes called, is implemented in a variety of ways in different computer programs. But by and large the rendering of surface shading usually includes the basic three components of local illumination surface reflectivity: ambient, diffuse, and specular. The basic three components of surface reflectivity usually refer to local **reflection of light**, and are also called **areas of illumination** of a surface. The effects of global scattering of light on the objects being shaded can be rendered with hybrid shading techniques such as ambient occlusion, or with global illumination or ray tracing, described in Chapter 6.

Different combinations of the surface reflectivity components can be used to simulate the surface characteristics of different materials. **Matte surfaces**, for example, can be simulated by using a combination of ambient and diffuse reflections. **Metallic surfaces** can be simulated with ambient and specular reflections. **Plastic surfaces** are typically simulated with a combination of ambient, diffuse, and specular reflections (Fig. 9.4.1). As described earlier, each one of these components can be rendered individually, as a separate rendering pass.

Ambient Reflectivity

The type of surface reflection that reacts to the intensity and color of the ambient light sources only is called **ambient reflectivity**. A unique characteristic of ambient reflection is that its intensity is independent of the distance between the reflective surface and the light source and also of the angle of the surface in respect to the light source. This means that light scatters evenly in all directions and that, as a result, all the polygons in three-dimensional models that are shaded with just ambient reflection end up with a uniform intensity and appear fairly flat, silhouetted sometimes. But when ambient reflectivity is used in conjunction with other types of reflectivity, it contributes to the overall intensity of the object. (Fig. 9.4.1). Because ambient reflectivity yields such a flat look, many practitioners and studios tend not to use ambient reflectivity at all, opting to use the technique of ambient occlusion (Fig. 9.1.1) or to simulate the ambient effect by raising the level of illumination with multiple lights that are shaded with diffuse reflectivity.

Diffuse Reflectivity

A surface with **diffuse reflectivity** reacts to incident light in different ways depending on the position and orientation of the light source in respect to the surface. Naturally, a surface with diffuse reflectivity will reflect more of a light source that is positioned next to it than a light source that is far away. But the most important factor in diffuse reflectivity is the angular position of the light source in relation to the object, more so than the distance between the light source and the object. Diffuse reflectivity is greater in areas of the surface that face the light source from a perpendicular angle. The

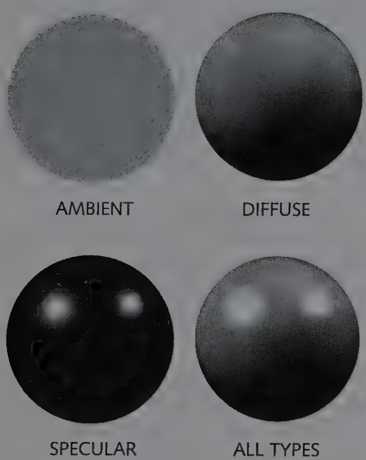
amount of light reflected with diffuse reflectivity decreases as the angle between the incident light source and the reflective surface becomes more oblique. Areas of a surface with diffuse reflectivity that are not reached by the light reflect very little or not at all (Fig. 9.4.1). The size of the surface area that faces the light source is also a factor in the intensity of the reflected light when using diffuse reflectivity.

9.3.14 This image from *Ernst Im Herbst* plays a reverse *trompe l'oeil* by bringing 3D back into a 2D world and making us wonder how this was put together. Sketches of the finished image were made and simple three-dimensional geometry, on which the characters cast shadows, built to match the drawn point of view. The three-dimensional sets were used as a reference for the hand-painted final backgrounds. The final paintings were colored in Photoshop using scanned textures. (Image courtesy of Studio Soi.)

Specular Reflectivity

Surfaces with **specular reflectivity** appear shiny because they reflect light the way a mirror does (Fig. 9.4.9). Specular reflectivity light does not scatter evenly throughout the surface. Instead, it is reflected in a focused and concentrated way, a characteristic known as **highlight sharpness**. In determining the amount of light reflected by surfaces with specular reflectivity, the position of the light source alone with respect to the surface is not as critical as it is in diffuse reflectivity. Instead, the apparent intensity of light reflected off surfaces with specular reflectivity depends mostly on the relation between the angle of the reflected light and the angle of the camera that is looking at the object. The intensity of the reflected light is greater when these two angles coincide. As the two angles move farther apart, the intensi-

9.3.15 An example of tiling using the spiral image map on Figure 9.7.1.



9.4.1 A sphere rendered with four different combinations of surface reflectivity. The first image is rendered with ambient reflection and an ambient light source. The second image is rendered with diffuse reflection only, one spotlight and one point light. The third image is rendered with specular reflection, one spotlight and one point light. The fourth image combines the three light sources and all types of surface reflectivity illustrated here.

ty of the reflected light decays sharply, a characteristic known as **highlight decay** (Fig. 9.4.1). Shading software programs provide accurate controls to vary the sharpness and decay of the specular reflectivity on surfaces.

Reflection Maps

Realistic reflection effects typical of shiny materials such as glasses, metals, plastics, or varnished surfaces are best obtained with ray tracing rendering. Ray tracing can simulate with precision the amount of reflections and refractions on a three-dimensional surface but it can also require much computation. A simpler strategy for creating reflective surfaces is based on the technique of reflection maps.

A **reflection map** consists of a two-dimensional image that is applied to a three-dimensional surface with the purpose of making the surface—or portions of it—reflective. A surface with a reflection map reflects the image of the three-dimensional models that are placed in front of the surface. The brightness values in a reflection map are used by the software to determine which parts of the surface are reflective and which are not. The dark or light values in a reflection map can be used to determine which parts of the object will be fully reflective. Reflection maps are usually monochromatic because the brightness values drive the simulation of reflectivity, and they can be projected onto three-dimensional surfaces with any of the standard projections used in texture mapping.

Environment Maps

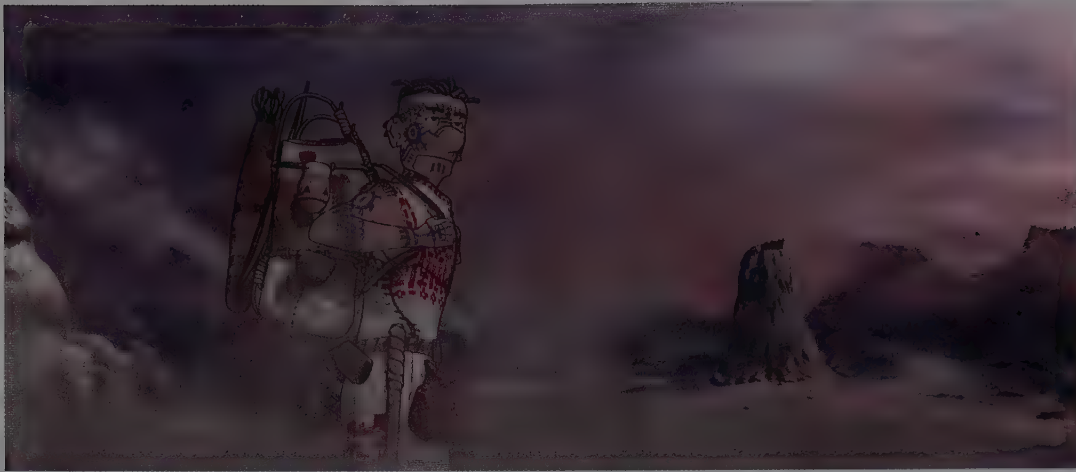
Environment maps can be thought of as a special type of reflection map because they reflect not only the objects surrounding the mapped object but also the environment surrounding the reflective surfaces (Fig. 9.4.2). The main characteristic of **environment maps** is that they are projected on all the objects with reflective surface characteristics in the scene and not just on one particular object. (Reflection maps are usually applied to one three-dimensional surface at a time.) The reflections of the surroundings on a group of objects can also be calculated with ray tracing rendering, but environment maps are often a cost-effective way to achieve similar results that are appropriate and sufficient in a large number of rendering projects. This technique is a favored alternative for creating the appearance of global reflections when ray tracing rendering methods are not used. When both ray tracing and environment mapping are active simultaneously, most rendering programs calculate the two parameters. However, in such cases, priority is usually given to the ray-traced reflections by placing them closer to the objects, and in front of the environment map reflections (Fig. 9.4.7).

Environment maps create an image of the area that surrounds the object as seen as if from the object itself. The appearance of the reflection of the environment is achieved by preparing a simplified

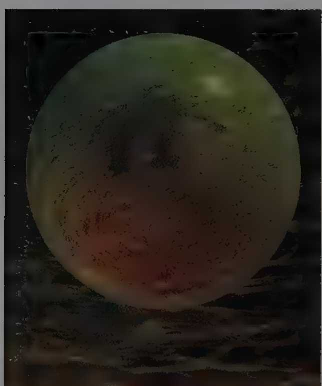
version of the three-dimensional environment in the form of two-dimensional images and then projecting those images onto the object with reflective surfaces as if they were the environment being reflected. Environment maps can be created with a variety of techniques. Two popular choices include a technique that resembles the spherical projection method and a technique that is an interesting variation of cubical projection of maps. In both cases, an image of the environment is first mapped inside the spherical or cubical space that contains the reflective object, and only then is the environment mapped onto the object or objects inside of the space (Fig. 9.4.4). Environment mapping works best when the maps are positioned far from the main characters or objects.

Spherical environment mapping is based on a flat image that is first projected on the inside of a sphere that represents the environment. The sphere is defined as a longitudinal space that goes from 0 to 360 degrees, and a latitudinal space that ranges from -90 to 90 degrees. Once the image representing the environment has been applied to the sphere it is then projected onto the object or objects that need environment mapping and that are placed inside the sphere. When an image is mapped inside a spherical environment, its left and right edges end up butting against each other, and its top and bottom edges are crimped. It is necessary to keep these mapping distortions in mind when preparing the image map for environmental

9.4.2 The textures on this *TMNT* character are a mix of procedural and painted textures rendered with RenderMan. The small circles in the chain mill that she wears under the armor were created with a displacement shader. (Teenage Mutant Ninja Turtles and TMNT are trademarks and copyrights of Mirage Studios, Inc. *TMNT* © 2007 Imagi Production Limited.)



9.4.3 The skin ornaments and textures of characters in *Morana* were hand-painted, and a cel shader was used to render with Lightwave software. A few layers of fog were generated with a particle system, each with the appropriate dimensions and tempo based on their depth, and later tweaked and composited. (© 2008 Kenges/Simon Bogojevic Narath.)



9.4.4 An environment map is mapped onto a sphere with 100 percent reflectivity, 90 percent specular highlights and two wave functions on its surface.

spherical mapping. It usually works best when the left and right edges of the image map match perfectly with one another; this way their projection can be seamless. It is also useful to keep the top and bottom areas of the image map uncluttered to avoid extreme distortion when the images on the top and bottom of the spherical space are somewhat compressed (Fig. 9.4.8).

The process of assembling an image suitable for **cubical environment mapping** is somewhat more demanding than preparing one for a spherical environment map. This is largely due to the fact that an environmental map based on the modified cubic projection is created by assembling six views of a scene. These six different views of a three-dimensional scene must represent a simplified view of the environment as if seen from inside an object that is placed at the center of this environment. The six views of the environment are the four side views, a top view, and a bottom view. The four side views are created by looking from the center of the environment toward the outside in angular increments of 90 degrees. In addition, each of the four side views must capture a full 90-degree view of the environment, so that when the four side views are assembled in sequence next to each other the result is a full 360-degree view of the environment (Fig. 9.4.10). The six panels required for a cubical environmental map can be painted, extracted from a real photograph, or created by rendering six different 90-degree views of a computer-generated three-dimensional space.

The images used in an environment map can also be generated with procedural techniques. Simple **color ramps**, for example, can be generated procedurally in the form of smooth gradations or blendings of color and are commonly used to represent clean skies or the chromatic effects of the sunrise or sunset on horizon lines. Clouds and an assortment of lighting effects can also be created procedurally and used as environment maps (Figs. 9.4.6, 9.4.11, and 12.3.4). Some programs provide rich procedures for generating envi-

9.4.5 The rendering is done with Mental Ray's Final Gather technique that calculates the diffuse reflections. In addition ray-traced soft shadows, four spotlights, and an area light were used. Three of the spotlights function as the key light, providing appropriate contrast and a soft shading transition, while the area light creates the soft shadows. The fourth spotlight acts as a subtle fill light. The diffuse reflections define a clear silhouette, eliminating the need for a rim light. Separate passes were rendered for color, shadow and ambient occlusion, and composited in Photoshop with the Blend Layer mode. The color layer was blured a bit to minimize sharpness but retain focus. See the wireframe geometry in Figure 4.5.2. (Image courtesy of David Tousek, Bohemian Multimedia.)

ronment textures of skies that take into account not only the position and brightness of the sun and whether the sky is cloudy or not, but also atmospheric parameters such as the curvature of the planet and the densities of the air and dust particles.

Environment maps can be animated to represent the motion that may happen around objects with mapped surfaces. The activity in a busy café can be simulated, for example, by mapping on a reflective sugar bowl a movie of people walking, drinking coffee, and interacting with each other. Another example of an animated environment map would consist of a movie of people dancing on the floor of a discotheque. This sequence could be mapped on the rotating silver sphere that is used to deflect light in all directions, or the eyeglasses worn by an observer.

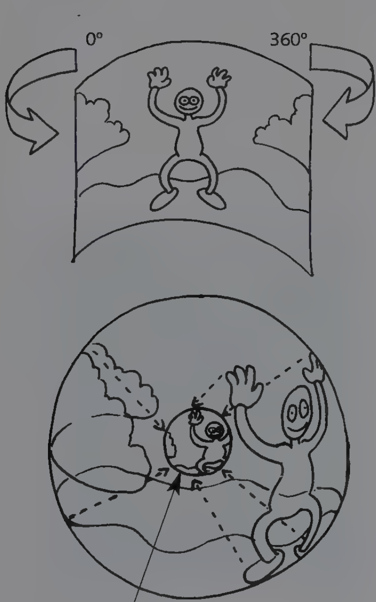
9.4.6 Detail of a still from the Danish feature *Terkel in Trouble*, showing the use of image mapping on the walls. (© 2004 A. Film A/S.)

Glow or Incandescence

Surface **glow**, sometimes called **incandescence**, is a surface characteristic that is associated with surface reflectivity. Incandescence makes objects glow in ways that resemble a variety of naturally

9.4.7 Specular surfaces with environment mapping in a scene from *Dragon Hunters*. (© MMVII Futurikon Films, Trixter, LuxAnimation, France3 Cinéma, RTL-Tvi, in coproduction with Mac Guff Ligne.)

9.4.9 (Opposite page, top) Carefully lit shot to highlight the craftsmanship in the Lamborghini Reventon engine, rendered in real time. (© RTT AG.)



9.4.8 In spherical environment mapping, a flattened image of the environment is created and then mapped inside a spherical space that contains the reflective objects.

glowing objects. Incandescent objects may appear as if they have an internal light source, like a fiber optic transmitting light, for example, or as if they generate a glow because of their extremely high temperature, such as molten lava (Fig. 9.4.12). This surface characteristic also resembles the opalescence of some gems that results from the reflection of iridescent light. Glow can be created as a uniform color across a surface or with an image map that determines which areas of the surface display glow. Glow can also be created with volumetric or lighting shaders, and through the skillful use of light sources in the three-dimensional scene.

9.5 Surface Color

Surface color is the most obvious of all surface characteristics. Color attributes are easier to identify and remember than most other surface characteristics. Surface color contributes to the personality of a three-dimensional character or the mood of a scene. When assigning a specific color to a three-dimensional surface it is important to keep in mind that the final color of the surface will also be greatly influenced by external factors such as the angle and color of the lighting applied to the surface, and even the color of the surrounding objects when rendering techniques such as radiosity or ray tracing are used.

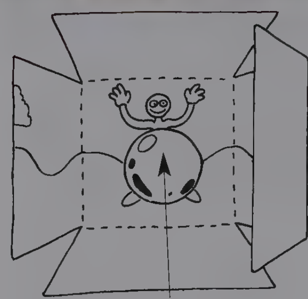
Surface color can be defined with a variety of color models, many of which are covered in Chapter 6. Additive or light-based color models, like RGB or HSL, are used to define the colors in images displayed on computer monitors. Subtractive or pigment-oriented systems, such as CMYK, are used to define the colors in printed media. When defining surface color, it is important to keep in mind the **color shifting** that occurs when a computer-generated image is translated from an additive color environment—like an RGB

monitor—to a subtractive color environment—like a printout on paper (see Chapter 15 for more details on minimizing color shifting).

Color Maps

There are different types of color maps and they all contain color information related to a three-dimensional object or scene. **Color maps** are used to compute the color of light reflected by the three-dimensional surface on which the color map has been placed (Figs. 9.5.1–9.5.3, and 9.3.11). A simple color map may contain just a single color value for each polygon in the scene, information which can be used by the shading program to calculate the color of each pixel after the light has reached that specific location. A color map may also store the RGB color values of each pixel after the color rendering pass has been shaded. This type of map with pixel color information can be used during compositing, to combine the color information in the scene with other rendering passes or layers such as a highlight or a reflection pass.

Color maps are sometimes called **picture maps** because they often contain images of paintings or photographs (Figs. 9.4.3 and 9.4.6). This type of color map can be projected and placed onto sur-



9.4.10 In cubic environment mapping, a six-panel view of the environment is created and then mapped inside a cube that surrounds the reflective objects. The small diagram indicates the orientation of each of the six two-dimensional views of the simulated environment.

faces by using any of the standard projection techniques described earlier in this chapter: flat, cubical, cylindrical, spherical, wrapping, and UV coordinates. Picture maps can be used in conjunction with other maps, such as displacement, transparency, and procedural textures to create a variety of effects and materials.

9.6 Surface Texture

It is possible to create interesting renderings by applying roughness onto the surfaces of three-dimensional objects. Modeling surface texture with true dimensionality, however, needs to be exercised with care as it might be appropriate only for some characters, scenes or projects. **Modeled surface texture** usually requires long rendering times. A common solution to this challenge consists of rendering the detailed geometry, turning the rendered image into a texture map and "baking" it into a simpler version of the geometry (Fig. 9.6.3). Texture mapping, also known as image mapping, is a simpler and more common technique for simulating surface texture and roughness. This method, pioneered in the mid-1970s by Ed Catmull, changes the intensity and chromatic values of a surface, but it does not affect its geometric smoothness.

The great variety of **texturing** techniques can be grouped in visual and spatial textures. **Visual textures** are flat simulations of three-dimensional texture and do not affect the geometrical surface of the object; they look textured but they are not. For example, a visual texture representing bricks is like brick **wallpaper**, and different from a real wall of bricks with relief textures that can be felt by touch. A practical benefit of using visual textures—in addition to their inherent aesthetic value—is that they make possible the creation of complex and rich textures with a minimal investment of polygons (Fig. 9.6.3). Some of the most useful visual textures include color and procedural maps, as well as environment, bump, and transparency maps. **Spatial textures** exist in three-dimensional space and affect the smoothness of surfaces. Spatial textures are closer than visual textures to the concept of a tactile texture. They usually require detailed meshes to be modeled with polygons and subdivision surfaces, with the fractal modeling techniques described in Chapter 5, or simulated with displacement maps.

9.4.11 (Above) Procedural generation of three-dimensional clouds, and image-mapped planes in *Blazing Angels 2 Secret Missions of WWII*. (Facing page, bottom) A model of the Gloster Meteor, the first jet fighter in the Royal Air Force. (Ubisoft Entertainment. All rights reserved. Blazing Angels, Ubisoft, and the Ubisoft logo are trademarks of Ubisoft Entertainment in the U.S. and/or other countries.)

Bump Maps

Bump mapping was developed by Jim Blinn and refined by a few others, to simulate roughness on a smooth surface not by affecting the surface itself but by altering the orientation of the surface normals. **Bump maps** provide an effective way to simulate roughness or bumpiness on a flat surface. Changing the orientation of the surface normals of polygons before shading causes the light to be reflected in several directions, simulating the way light would be reflected from an object with rough surfaces. This results in the appearance of a tex-

9.5.1 Color map with the skin tones applied to the virtual Marlene Dietrich in Figure 9.3.3. This color map was created with a combination of painting and scanning techniques. The lips, for example, were captured by "printing a kiss" with lipstick and scanning it. (© 1999 Virtual Celebrity/Marlene Inc.)

tured surface with modulations that resemble the pattern contained in the image file used as a bump map (Figs. 9.3.1 and 9.6.1). The darkest values in the image map may represent the valleys, and the lightest values simulate the peaks in the simulated texture, or vice versa.

Bump mapping has a few limitations, however, when used by itself. A geometric surface with bump maps, and its edges in particular, remain flat. For this reason objects mapped exclusively with bump maps should not be placed too close to the camera. This flaw is accentuated when the bump map has a wide range of brightness values that create the impression of increased texture and that may contrast a lot with the smooth profile of the object. In addition the bump map-simulated peaks do not project shadows.

Bump maps are often used in combination with parametric waves to represent the motion of water. This motion is usually limited to the linear or concentric undulations created on the surface of the water by wind or objects that touch a point on the surface (Fig. 11.2.8). Parametric waves used as bump maps can also be used as displacement maps to create a more three-dimensional effect.

9.4.12 These images show the effects of shading the incandescent object in the center with (top) and without (bottom) a surface glow.

Displacement Maps

Displacement maps provide a unique way to use an image map to modify the shading of the surface and its geometry. **Displacement**

maps change both the orientation of the surface normals and the three-dimensional position of the surface itself. This results in a surface with displaced vertices that can also have two-dimensional patterns mapped on it (Figs. 9.3.1, 9.4.2, and 9.6.2).

Displacement maps are often used to create three-dimensional terrain that includes mountains and valleys. Terrains can be built with displacement maps that are based on photographic images of aerial views where the different elevations are coded with different colors or shades of gray. Complex three-dimensional terrains can also be easily built by generating two-dimensional fractal images and using them as displacement maps. Whether photographic or fractal, the two-dimensional images are usually applied to a three-dimensional surface in the form of a black-and-white displacement map and as a color image map.

Two-Dimensional Procedural Texture Maps

As mentioned earlier, the two-dimensional images that can be mapped onto three-dimensional surfaces can be painted by hand, captured with cameras, or created with procedural techniques. **Procedural creation** relies on mathematical functions or computer programs to create images that are usually abstract. Mathematical functions that create pseudo-random or rhythmic patterns of color are popular ways to create two-dimensional procedural texture maps (Fig. 9.6.4). Two-dimensional images that are created with procedural techniques can be mapped onto three-dimensional surfaces following the standard procedures for image mapping (Fig. 9.4.2).

Three-Dimensional Procedural Texture Maps

Many of the textures found in nature can be easily simulated with three-dimensional procedural texture maps. These are **solid textures** that exist on the surface of an object and extend inside and throughout the entire object. These textures are based on mathematical functions or short programs that create abstract patterns. But unlike two-dimensional procedural textures that are projected onto the surface of an object, three-dimensional procedural texture maps create the three-dimensional patterns throughout the object being textured. A small chunk of marble is a good example of how three-dimensional solid textures behave. On the outside, marble has a very distinct texture defined by the colors and surface characteristics of the minerals that it is composed of. But the texture of marble does not occur only on its surface—it continues inside of the stone because the minerals that define the marble are all through the stone. The texture of marble cannot be peeled off like a two-dimensional texture map could. When the stone is chipped or cut, the surface texture changes, revealing the inner mineral composition and continued texture of the stone.

Many natural and synthetic materials can be recreated with procedural solid textures. This is done by providing different values to a

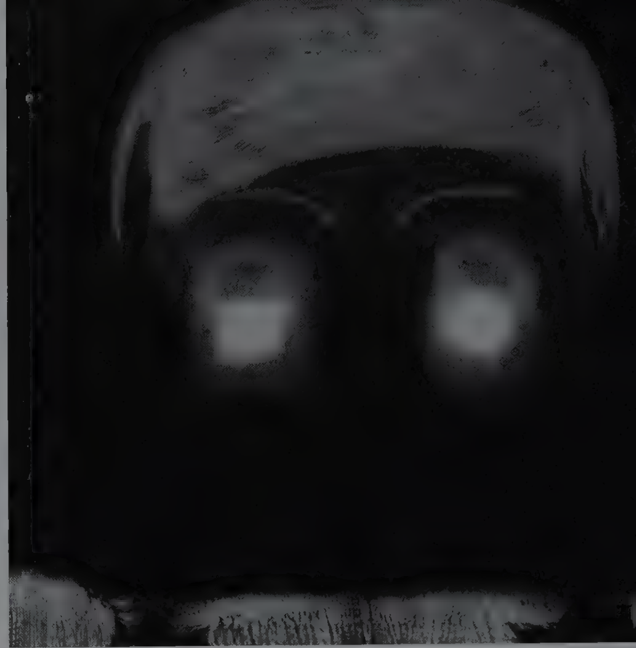
9.5.2 Scenes with about 40,000 polygons each from *Happy Tree Friends–False Alarm*. The flat shading emphasizes surface color and simple image maps. Rendered with a custom graphics engine using hardware acceleration via DirectX 9. (© 2008 Mondo Media. All rights reserved.)

variety of procedural parameters, such as color, roughness, frequency, scaling, orientation, cohesion, and density. These values can be typed directly on the keyboard or controlled interactively with sliders. Some natural materials that are offered as prepackaged standard options in many software programs include a variety of stones, such as marble and granite, and also wood, corroded metal, leather, and even smoke and clouds (Fig. 9.6.5).

9.5.3 Hand-painted color maps applied onto cutout figures and billboards. (Created and directed by Wolf-Rüdiger Bloss. © Tube Caveman Inc.–Antefilms Production.)

Software programs have slightly different ways for defining three-dimensional solid textures, and each produces a unique result (Fig. 1.2.4). Marble textures, for example, can be defined with variables Color 1, Color 2, X Weight, Y Weight, Z Weight, Turbulence, and Cohesion. Marble can also be defined with variables Filler_color, Vein_color, Vein_width, Diffusion, and Contrast. Other general three-dimensional solid texture variables associated with marble definitions include some noise parameters and the minimum and maximum levels of recursion depth.

Many procedural texturing techniques are based on the idea that some degree of controlled randomness is useful, even necessary, to



9.6.1 The monochrome image on top left was used as a bump map to render the pores of the skin. To capture the map, ink was rolled on faces and an imprint was made on paper. The eyelashes were created with an additional bump map not shown here. The darker map (top right) was used to accentuate the specular highlights during rendering. The small color image map was used for the right ear of the character in Figure 9.3.3. (© 1999 Virtual Celebrity/Marlene Inc.)

define the characteristics of specific textures. This randomness is often specified in the form of a **noise** function that generates **stochastic** (or pseudo-random) values, and feeds them to the parameters in the texture procedure. One way to define the turbulence in marble-like surfaces, for example, is with stochastic patterns whose magnitude decreases with frequency. This means that as the marble texture patterns become tighter—as their frequency increases—their area on the surface becomes smaller.

Procedural textures can be offset across the model that they are applied to (Fig. 9.6.9). Procedural solid texture maps do not use standard projection techniques. Solid textures exist throughout the object (outside and inside), and there is no need to project them.

9.7 Surface Transparency

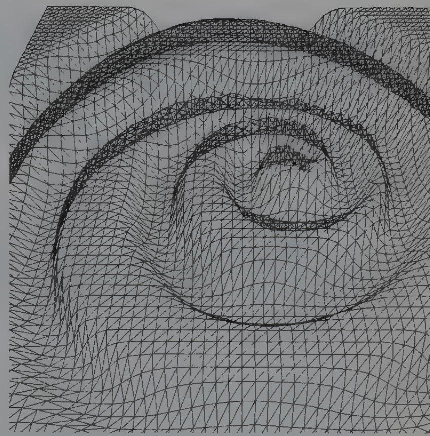
Transparency and translucency effects are useful for rendering materials such as glass or water, and also for visualizing fantastic transformations of matter—for example, opaque charcoal turning into a transparent diamond. **Surface transparency** is represented by simulating the behavior of light on transparent materials. Realistic transparency effects are best simulated with the ray tracing rendering method (described in Chapter 6), which usually provides accurate controls for transparency and for light refraction.

Transparency Maps

Another strategy for simulating transparency consists of applying transparency maps to the surface of a three-dimensional object. A **transparency map** consists of a monochromatic two-dimensional

image that is applied to a three-dimensional surface with the purpose of making all or some of it transparent (Fig. 9.3.13). The basic idea behind a transparency map is that the rendering program looks at the **brightness values** of the pixels in the map and uses them to determine whether the surface will be transparent, opaque, or translucent.

Some programs use the black values in the transparency map as the indicator for full transparency (Fig. 9.7.1), while other programs use the white values to activate full transparency. In either case, a gray transparency map results in a translucent surface. Creating a transparency map in full color is a waste of time because the majority of shading programs only pay attention to the grayscale brightness values when dealing with transparency maps.When transparent surfaces are rendered with both reflection maps and reflectivity settings in the ray-tracing rendering method, the ray tracing of reflected three-dimensional objects takes precedence over the reflection mapping.



9.6.2 A terrain created by displacing the XYZ locations of a surface with the black-and-white image map shown in Figure 9.7.1. This technique can be used to animate the effect of footsteps left on the ground by a walking ghost by using a sequence of footprints (one at a time and in sequence) to displace the ground down.

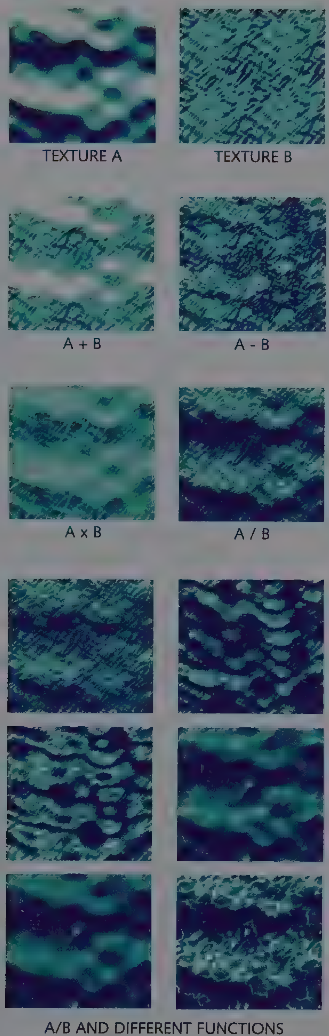
9.8 Environment-Dependent Shading

A large variety of shading attributes are determined by the characteristics of the three-dimensional environment in which the rendered objects are placed. Some of the most common tools for controlling environment-dependent variables include antialiasing, motion blur, and depth-fading. Some aspects of environment-dependent shading can also be controlled through volume shaders, a type that is native to the RenderMan environment and goes beyond surface shaders. **Volume shaders** define the characteristics of materials in three-dimensional space that affect light as it travels through them. Our atmosphere, for example, contains gases and solid and liquid particles that affect light before it reaches our eyes and after being emitted by light sources or being reflected by surfaces. The characteristics of the light or imaging rays that travel through a volume—for example underwater scenes or the inside of the human body (Figs. 5.1.6 and 8.1.7)—can be defined based on attributes of the volume, such as density, color, and motion.

9.6.3 The level boss Balrog with an assortment of surface textures, some of them are renders "baked" as image maps. (*Kingdom Under Fire: Circle of Doom*™. Reprinted with permission from Microsoft Corporation.)

Antialiasing

When the spatial resolution of an image is too low, its details are often lost. This phenomenon is called **spatial aliasing**, and it occurs when the details in the image are smaller than the size of the individual pixels used to represent the image. Aliasing, as spatial aliasing is usually referred to, is usually seen in the form of **jagged edges**, especially in objects with diagonal and curved profiles. Aliasing effects can be found not only in computer generated images but in images generated with other media, such as painting or photography, for example, when the brush strokes or the film grain are larger than the image details. Aliasing can also be thought of as the image distortions that result from a limited or insufficient sampling of the original visual data.



9.6.4 Examples of procedural two-dimensional textures. The first two textures from top left to bottom right were generated with procedures based on functions with different levels of complexity, detail, twisting, and contrast. The next four textures were created by blending the two original textures with the same function curves, and the following operators: A + B, A - B, A × B, and A/B. The remainder of the textures were created with the A/B operator, and a different combination of function curves each time.

The best way to eliminate aliasing is to increase the **spatial resolution** of an image, which means to increase the number of pixels in the image. This will also increase the time that is required to render the three-dimensional scene, because the number of rendering calculations is related to the total number of pixels in an image. Alternative methods for eliminating aliasing effects are called **antialiasing**, which is usually based on **oversampling** and **interpolation** techniques. These techniques determine the color value of a pixel by first examining the value of the surrounding pixels, then averaging those values, and finally using that average to determine the value of the individual pixel.

There are many antialiasing algorithms, and some are more efficient and accurate than others. Some antialiasing techniques can dramatically increase the quality of an image but sometimes at the expense of performance. For this reason, when choosing degrees of antialiasing, those responsible for rendering must consider the essential factors in a specific production—such as deadlines, budget, and quality desired—and apply their best judgment (Fig. 9.8.1).

Subsurface Scattering

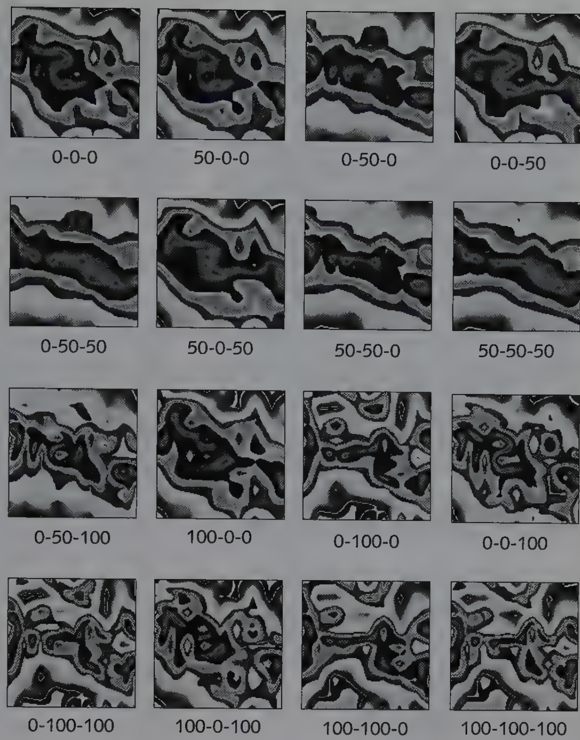
Subsurface scattering techniques have the ability to calculate the behavior of light as it travels back and forth through translucent surfaces and layers. This technique takes into account the multiple layers of skin, the body fat, and the blood vessels. Renderings of human skin with subsurface scattering techniques are convincing (Figs. 1.2.9, 8.1.10, 9.1.1, and 12.1.10). The rendering of subsurface scattering effects is possible with a variation of the BRDF local illumination light reflectance model called the BSSRDF, or bidirectional subsurface scattering reflectance distribution function.

Motion Blur

When recording reality with a video or a film camera we observe blurred objects when they move too fast in front of the camera. This phenomenon is called **motion blur,** and it occurs naturally in film or video recordings where the shutter speed is too slow to freeze an object in motion. Motion blur is a form of **temporal aliasing** that results from samples that are too far apart to capture motion details. The speed of shutters in photographic still cameras is measured in the amount of seconds or fractions of a second that the camera shutter remains open. Usually, speeds of 1/250th of a second are necessary to freeze fast-moving objects. The speed of shutters in motion film or video cameras is usually measured in terms of the number of frames that are recorded per second. Most motion cameras have fixed speeds of 24 frames per second (fps) for 35 mm film, and 30 fps for video. Only high-speed motion cameras are capable of high shutter speeds that can freeze the motion of objects. This is achieved by recording a large number of frames per second and, therefore,

slowing down the motion of the objects.

Motion blur can add a touch of realism to computer animation because it reminds viewers of the blurring effect that occurs when we record fast-moving real objects directly with a camera (Figs. 9.6.7, 9.8.3, and 9.8.5). But motion blur must be added to computer animation, since it does not occur naturally in it, and there are a few methods to do this. Motion blur can be defined by specifying a shutter speed expressed in seconds or frames per second, and also the rate at which moving objects are sampled while the shutter is open. For animation that is recorded at 30 fps, for example, a camera shutter that remains open for two frames has a speed of 1/15th (or 2/30ths) of a second. Motion blur can also be calculated based on a minimum number of pixels that the object moves within the two-dimensional space of the image plane of the camera (Fig. 9.8.4). In many cases motion blur is applied after the initial calculation of the position of the three-dimensional object in the three-dimensional scene. Motion blur is also often applied as a two-dimensional effect after the rendering and shading process is finalized.



9.6.5 Variations of a solid wood texture cross section created with different values of swirling, grain density, and cutting. These previews are used to quickly visualize how a solid texture looks before it is sent to a detailed rendering. Examine the first example with values of 0-0-0, the tenth example with maximum swirl values of 100-0-0, the eleventh example with maximum grain values of 0-100-0, the twelfth example with maximum cutting values of 0-0-100, and the sixteenth with overall maximum values of 100-100-100. See Figure 9.6.9 for another texture created on a cylinder using these four variables.

Fog and Depth-Fading

Most rendering software provides atmospheric or environmental shading tools for simulating the effect of **fog** in a three-dimensional scene. The presence of fog makes the three-dimensional images fade into the color of the fog according to their position relative to the camera (Figs. 9.8.6 and 9.8.7). Objects that are far away from the camera and deep into the three-dimensional scene blend more with the fog. For this reason this technique is sometimes called **depth-fading**. In most cases, fog and depth-fading are just two different names given to the same function, but in some programs, fog and depth-fading are two different functions.

There are many algorithms for calculating fog, and some produce more realistic effects than others. In general, the functionalities of fog tools control the starting and ending distance of the fog, its color, and sometimes its transparency. The starting and ending distances—also called minimum and maximum distances—of the fog specify the distances from the camera to the plane where the fog starts and ends. Objects that are closest to the ending distance blend more with the fog. The color of the fog can be specified with a variety of color models, and in many cases the color of the fog is related to—or determined by—the **background color**, a global shading parameter. The transparency of the fog defines the degree to which

9.6.6 *Smile* displays a rich variety of surface shading and modeling techniques to create a hopeful moment in a dark space. (© Alex Alvarez.)

9.6.7 The foreground soldier (left) was rendered with motion blur as a shrieking wretch attacks. (*Gears of War 2*™ Epic Games, Inc. Reprinted with permission from Epic Games, Inc.)

the objects placed behind the fog are visible. Opaque fogs, for example, may block everything that is placed behind them, while somewhat transparent fogs allow the objects behind them to partially show through. Many fog functions provide a **visibility** parameter that is usually related to the depth, or thickness, of the foggy area. Visibility can be easily determined by subtracting the starting distance of the fog from the ending distance.

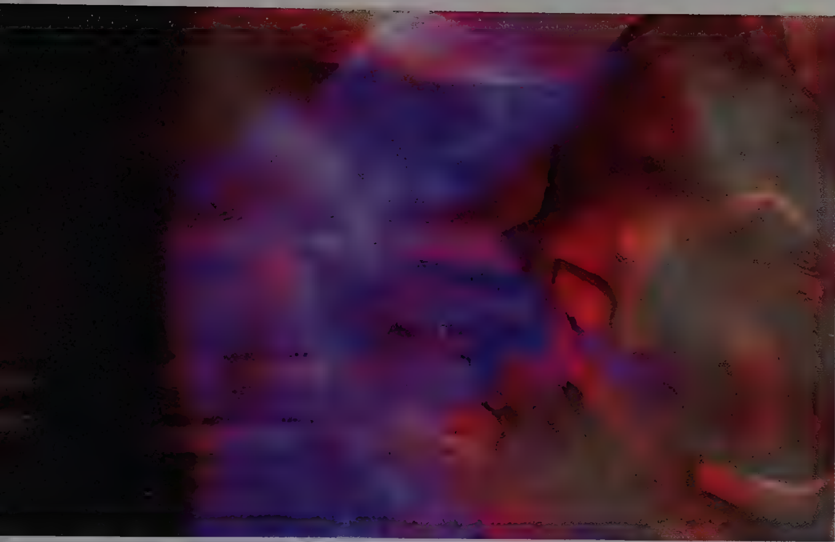
Both fog and depth-fading techniques can be used to create images that incorporate the principles of aerial perspective refined by Leonardo da Vinci in the sixteenth century. **Aerial perspective** techniques are used to depict depth in a two-dimensional image by simulating the atmospheric effects of light, temperature, and humidity on objects situated far away from the observer. Aerial perspective was developed to complement the realism of images represented with the principles of linear perspective. (Linear perspective techniques are implemented in the determination of visible surfaces that occurs during the rendering process.)

Many software programs provide fog and depth-fading functionalities, but often these two words represent different versions of an effect. This is the case, for example, with software programs that support the functionalities provided by the RenderMan programming language. The difference between fog and depth-fading basically lies in the way in which each of these operators adds the background color to the light that is reflected by three-dimensional surfaces. Both operators add the background color to the reflected light based on the distance between the surface and the position and orientation of the camera. But in depth-fading, the color of a surface is entirely the background color if the surface is beyond the maximum distance, while in fog the reflected light always retains some of the color originally reflected by the surface.

Fog, clouds, and smoke can also be simulated as a cluster of particles with uniform density that is placed in front of the camera (Figs. 9.4.3 and 6.10.9). Small clusters of fog, smoke, or steam can be created with the solid procedural textures described earlier in this chapter. In addition to fog that extends only horizontally, some programs offer the capability of simulating distant layers of fog placed at different altitudes. This is usually achieved with an image map that has a painting or a photograph of the horizontal layers of fog. This image map is used to calculate the color of light reflected by objects behind the fog plane or by a color map that is used as a backdrop. Another shortcut to rendering fog consists of mapping painted foggy scenes on a flat billboard with varying degrees of translucency.

9.9 Getting Ready

The rendering process is finished only when all the layers and elements in a scene have been assembled into a single frame that includes all the changes and fixes that were specified throughout the



9.6.8 False color rendering and compositing were used to create the "spirit" phantoms in *Final Fantasy: The Spirits Within,* a show rendered primarily with RenderMan, and Maya for ray-traced effects such as caustics. Photoshop and Amazon Paint were used to create matte paintings and textures. The modeling and animation were done mostly with Maya, and the compositing with Illusion, Shake and Flame. (© 2001 FFFP.)

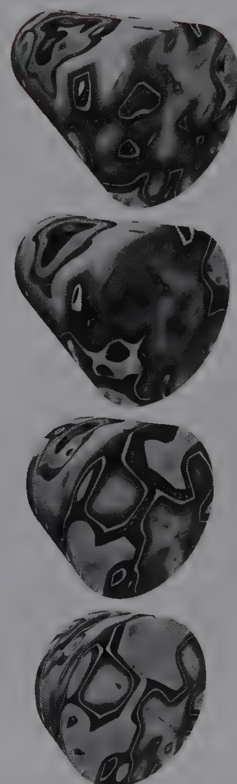
review process. Oftentimes in a project, an individual or group of individuals are focused on collecting the rendered layers, and adding any last-minute rendering that might be missing or that might need to be revised based on the director's comments. These technical tasks are commonly called **rendering finaling** or **rendering wrangling**, and are closely related to compositing (Fig. 9.1.1).

Selected Rendering Hacks

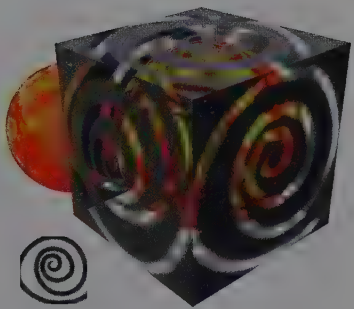
Hacks or **cheats** are shortcuts that yield results similar to the established technique with less work, time, and/or computing power. A hack is sometimes an effective way to develop a one-time solution that may be quicker and cheaper than other more established or technically solid methods. The use of rendering hacks in a production pipeline should be carefully considered as they tend to be locked into a specific approach and making changes can sometimes be difficult and/or time-consuming.

Hacks are sometimes used extensively during the rendering process because they can be a great help in reducing the computing time that a complex rendering setup might require. Interestingly enough, many hacks are based on nonstandard tricks that eventually spawn techniques that become standard practice. Ambient occlusion, for example, is a simplification of the global illumination rendering method but is used extensively because it is faster and yields good-looking results.

Billboards are widely used to insert in a scene image maps of more complex scenes that have been prerendered or recorded from live action. Billboards are usually flat polygons that are mapped with a still or a sequence of image maps. In real-time games, for example, billboards are great for mapping prerendered animated sequences of



9.6.9 Slicing a cylinder reveals how solid textures exist throughout objects. See Figure 15.2.2 for a lower resolution version.



9.7.1 Transparency map applied to a cube with cubical projection, using the black and white pattern from Figure 9.6.2.

9.8.1 The same detail of a three-dimensional scene rendered with and without antialiasing. Notice the significant differences in edge jaggedness in the steep diagonal edges, and color blending especially around yellow.

distant characters. When used effectively this hack can look as good as having the full three-dimensional character in three-dimensional space: one polygon instead of hundreds (Figs. 4.7.7–4.7.9, 6.11.1, 9.5.3, and 11.6.3). In live action feature film, billboards are sometimes used to map live footage in an all-synthetic shot. That is the case, for example, with the actor inside the helicopter in ***Mission Impossible***'s tunnel sequence. In either case, the images are mapped on the billboard with a transparency channel, to make transparent the billboard areas that do not have an image on them. Billboards are also often used to hold single stills of distant backgrounds that have been photographed on location or from scale models (as was the case in the pod race sequence in ***The Phantom Menace***). Earlier in this chapter we mentioned the popular approach of applying procedural textures rendered on a flat plane as image maps to other geometry. This greatly saves rendering time.

Selective ray tracing minimizes rendering times without impacting the quality of the visual end result. In a crowd scene, for example, only those closest to the camera can be ray traced, while the ones far away are rendered with other simpler and faster techniques such as reflection and environment maps. This was the case in several shots with the little green creatures in the movie ***Flubber***.

Simulated film grain was achieved very successfully in the movie ***Bunny*** by computing just a couple of iterations in the radiosity rendering cycle. While this may not be the cheapest way to simulate film grain in computer-generated images, it creates a beautiful and unexpected visual quality by using an established technique for doing things that it was not intended to do. ***Bunny*** is also an example of selective ray tracing since only the furry characters were ray traced while the environments were rendered with radiosity.

Simulated lens glows and flares can be quickly inserted in a three-dimensional environment by mapping preexisting images of glows and flares on a polygon that overlaps with the near clipping plane of the image. More orthodox glows and flares are achieved by computing the diffraction and refraction of light rays inside the optics of the lens, but the computing overhead can be significant (Fig. 10.1.8). Using stills or sequences of photographic or prerendered glows and flares saves time since the main rendering challenge in this hack becomes to line them up with the scene and to blend them using a delicate transparency map with soft edges.

Rendering in a Network

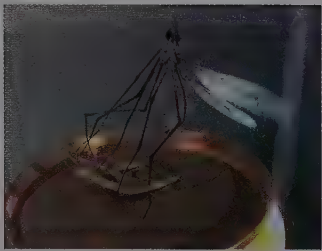
Computer networks can be used to increase the speed of your renderings. But keep in mind that the final performance depends on the specifics of the network as much as the network rendering features of your software and computer. The two most common strategies for sending your rendering to other machines on the network are distributed rendering and remote rendering. **Distributed rendering** consists of sending portions of a rendering job to different computers on

the network—for example, rendering the top half of a scene on one machine and the bottom half on another. Distributed rendering requires software that is able to split a rendering job into several sections and then put the results back together. It also requires the permission to access the machine by its owner. **Remote rendering** occurs when the rendering of a three-dimensional model—that may reside in your machine—takes place in one of the other machines on the network. Many companies today have **rendering farms** that consist of many computers solely dedicated to network rendering (Fig. 2.5.3). These rendering farms may be located in the same building as the rest of the company, or they may be located in other buildings, other cities, or in other countries, where rendering labor may be more economical. (See Chapter 2 for more details on rendering farms.)

9.8.2 The lighting in this simulation of a womb highlights the shading characteristics of subsurface scattering techniques that showcase the multiple layers of skin, fat, and capillaries under the skin surface. (Image courtesy of The Mill.)

Streamline Your Shading Data

Shading parameters, such as the number of surface layers or the number of image maps applied to surfaces, should be kept to a minimum, especially in productions where rendering power is limited or rendering time is of the essence. Very often, the best rendering results are achieved with just a few well-chosen shading parameters. Too many shading parameters not only prolong the time needed to render a scene but also might not significantly contribute to the



9.8.3 The effects of motion blur can be seen on the wings of this mosquito who is trying to open a beer can. Since the camera is stationary the story in this scene is carried only by the mosquito's performance. (Courtesy of Blue Sky Studios. © 1998 Tennent Caledonian Breweries, Inc.)

9.8.4 The motion blur of an object increases with its speed and as the object gets closer to the camera.

visual quality of the final image. It is the responsibility of the production team to find a balance between the essential aesthetic needs and the practical limitations of the project, such as the rendering speed of the computer system, peoples' schedules, the budget, and the delivery deadlines. However, highly realistic visual effects that match the live action are oftentimes a production requirement.

Rendering Glass

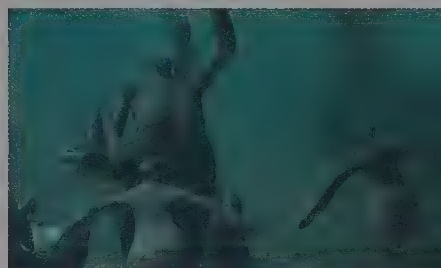
Many rendering programs provide users with predefined shading parameters or surface shaders for a variety of materials, including glass (Figs. 6.6.2, 6.6.4, and 12.3.7). Depending on the color, thickness, transparency, and roughness of the glass being simulated, both the reflectivity and the transparency of the object should be set high—values above 90 percent are not uncommon. The refractive index can be set slightly above normal to create a small amount of distortion, or it can be boosted to simulate the increased light-bending qualities of handmade glass, for example. Specular highlight values can be sharpened and focused, and specular reflectivity can be boosted up above 100 percent to make sure that the highlights are visible in highly reflective and transparent glass. Try turning off ambient and diffuse reflectivity altogether. The transparency and reflection depths of a ray-traced rendering must be set to a bare minimum value of 4 in order to capture the subtle distortions that give refraction its tantalizing qualities.

Keep Digital Backups

The preferred method for recording computer animations consists of accumulating a number of rendered still frames in peripheral memo-

9.8.5 Motion blur enhances the illusion of fast-twirling hair in this detail of a shot from *Spy Kids 2*. (© 2002 Hybride. Images courtesy of Dimension Films.)

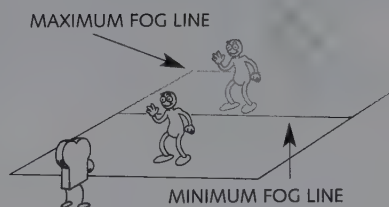
ry and then recording them onto film or videotape. This batch mode of recording animation is usually more efficient than recording still frames one at a time as each is rendered. It is convenient to keep all the files of the digital still frames, even after they have been recorded on film or videotape. Keeping a **digital backup** of the animation is invaluable if anything should happen to the master videotape or the original film negative before the project is delivered. Discarding the computer animation still frames as soon as they are recorded onto film or videotape is a situation that should be avoided at all costs. If this were done and the master videotape or original film negative were damaged, or the client unexpectedly asked for delivery in a different format, then large portions of the project would have to be rendered all over again.



9.8.6 A still from *How Mermaids Breed* illustrates the use of depth fading to represent the effect of underwater depth of field. (© 2002 Joan Ashworth.)

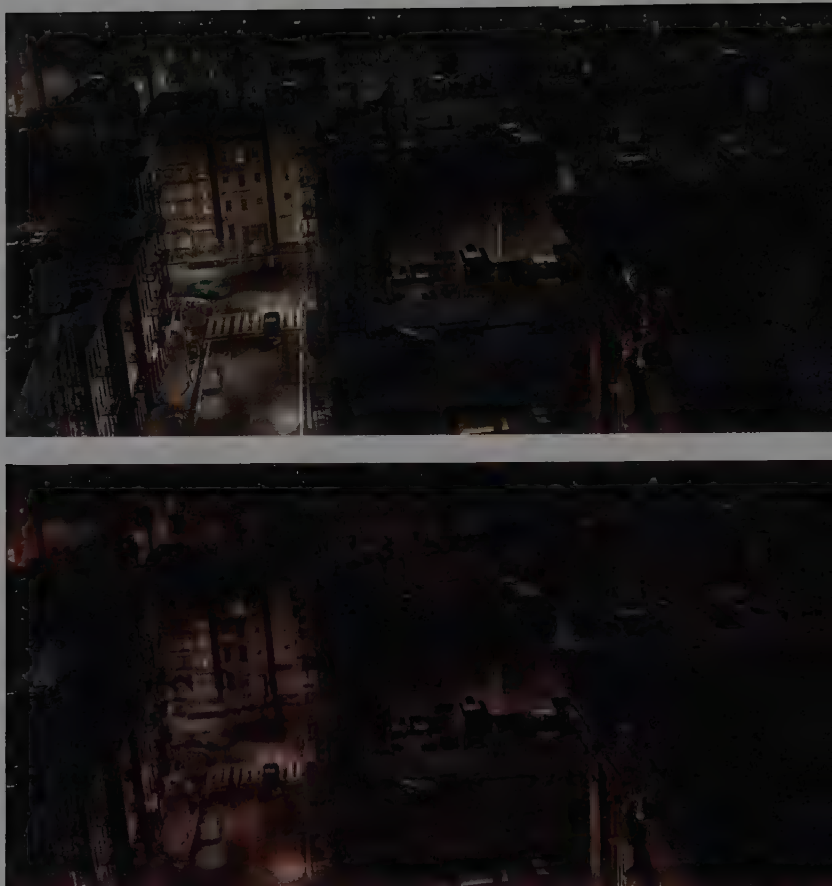
Consider the Final Output Media

Before producing your final renderings, consider for a moment what the final output media is. As mentioned in Chapter 6, there can be significant color shifts when recording an RGB computer-generated image onto media with other chromatic ranges. This becomes especially critical when recording still images or animated sequences onto videotape due to the medium's narrower range of color as compared to RGB. Special care should also be taken to make sure that the computer-generated images fall within the chromatic range of film—for example, when the computer animated characters are composited over a live action background plate (Fig. 14.1.1).



9.8.7 The shading of objects within the foggy area in this diagram gets influenced by the color of the fog.

9.9.1 The changes between the two city night shots are lighting and color tweaks done in compositing. The original rendering (top) was done with RenderMan, and ambient occlusion was not used. (*Teenage Mutant Ninja Turtles* and TMNT are trademarks and copyrights of Mirage Studios, Inc. *TMNT* © 2007 Imagi Production Ltd.)



9.9.2 Two lighting versions of same shot following the director's comments. (© 2007 Imagi Production Limited. Credits same as above.)

9.9.3 Shifu uses proven teaching techniques to make Po realize that he is a kung fu master. Notice the clear silhouettes, telling body language, and easy-to-read facial expressions. (*Kung Fu Panda*™ and © 2008 DreamWorks Animation LLC, used with permission.)

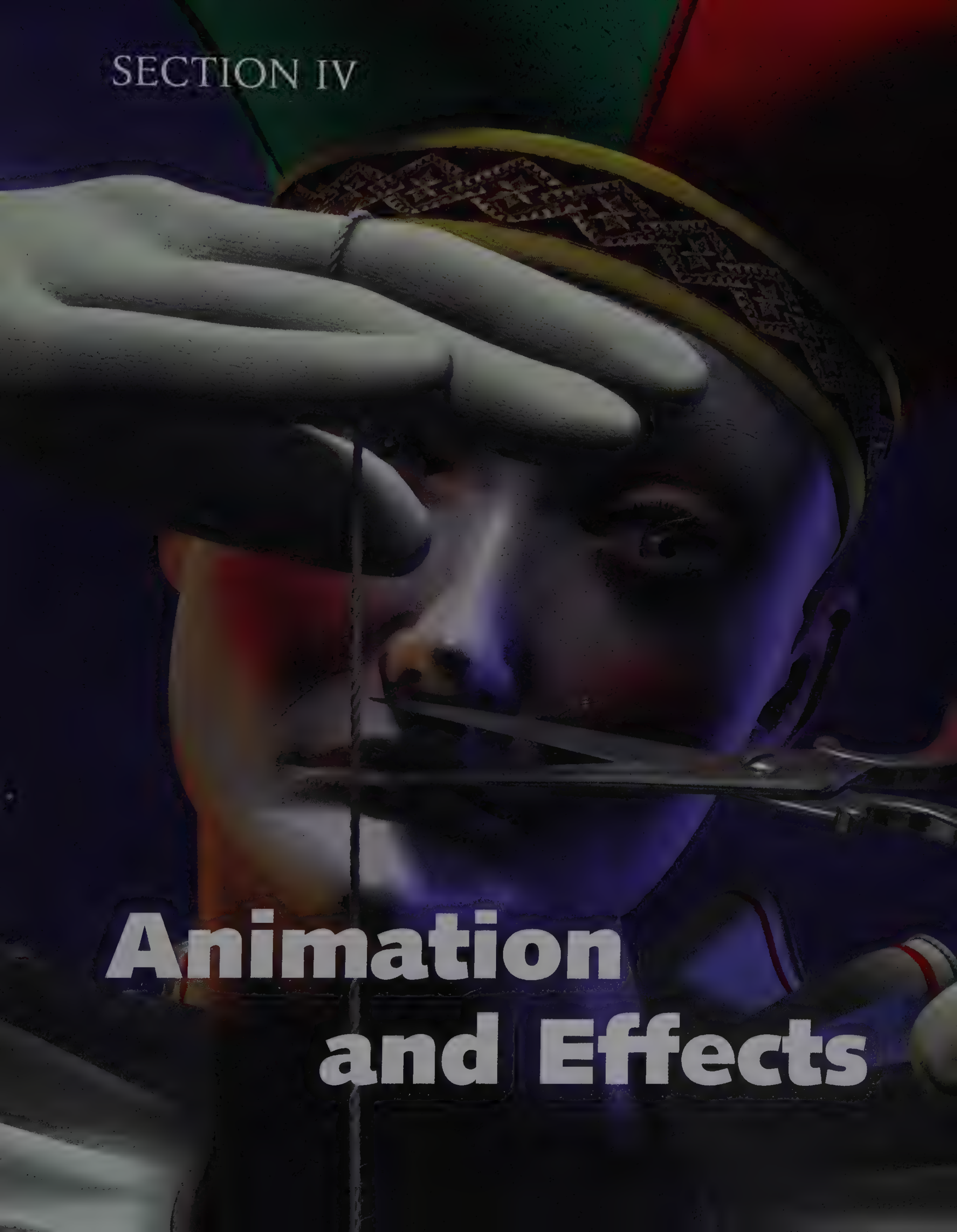
CHAPTER 9

Key Terms

Aerial perspective
Alpha channel
Ambient reflectivity
Ambient occlusion
Angle of mapping
Antialiasing
Areas of illumination
Averaging
Backdrops
Background color
Billboards
BRDF, Bidirectional Scattering Distribution Function
Brightness values
Bump maps
Cheats
Color maps
Color ramps
Color shifting
Continuous masks
Continuous visual information
Cubical environment mapping
Cubical projection
Cylindrical projection
Depth-fading
Diffuse reflectivity
Diffuse surface shading
Digital camera
Digital painting
Digital scanner
Dioramas
Discrete numerical values
Displacement maps
Distributed rendering
Environment maps
Faceted appearance
Flat projection
Fog
Glow
Gouraud shading model
Hacks
High-contrast masks
Highlight decay
Highlight sharpness
Hybrid shading models
Image mapping
Incandescence
Interpolation
Jagged edges
Lambert shading model
Light reflectance model
Local illumination
Map blending
Mapping projection method
Mask
Material databases
Matte surfaces
Matting techniques
Metallic surfaces
Mirror-like
Modeled surface textures
Motion blur
Noise
Numerical values
Offset the map
Overall blending
Oversampling
Parameter space
Phong shading model
Photorealistic RenderMan
Picture maps
Plastic surfaces
Polygonal shading
Pressure-sensitive
PRMan
Procedural creation
Rectangular images
Reflection map
Reflection of light
Renderer
Rendering farms
Rendering finaling, wrangling
Rendering layer, pass
RenderMan shading language
Remote rendering
RIB, Renderman Interface Bytestream
Sampling
Scaling
Selective ray-tracing
Shading value
Shadow maps
Simulated film grain
Simulated lens glows and flares
Simulated material
Smooth surface shading
Solid textures
Spatial aliasing
Spatial resolution
Spatial textures
Specular reflectivity
Specular surface shading
Spherical environment mapping
Spherical projection
Stochastic
Surface finish
Surface illumination
Surface layers
Surface libraries
Surface normals
Surface shader
Subsurface scattering, SSS
Surface transparency
Temporal aliasing
Texturing
Tiling
Transparency map
Underpaint
UV coordinates
Vertex normals
Visibility
Visual textures
Volume shaders
Wallpaper
Wrapping
XY coordinates

SECTION IV

Animation and Effects



(Previous page) The protagonist in *Tightrope*. (© Digital Domain, Inc.)

10.1.1 A minor event in the life of *Bunny*'s main character makes her confront her memories and her future. (© 1998 Blue Sky Studios.)

CHAPTER 10

Principles of Animation

QUICK INDEX

Summary

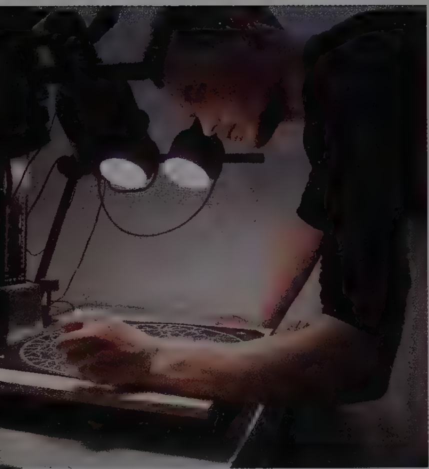
USING COMPUTERS TO CREATE ANIMATED IMAGES offers animators new creative possibilities as well as potent production tools. This chapter reviews some of the basic concepts of animation, including fundamental techniques such as keyframing and in-betweening, the communication of emotions and thought processes through an animated character, and the use of storyboards to present your ideas.

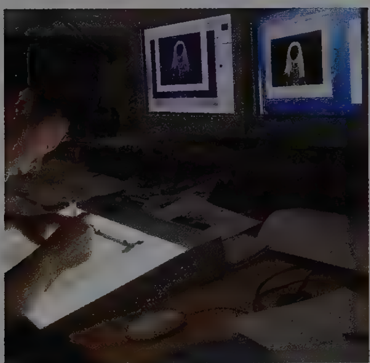
10.1 The Craft of Animation

To animate means to give life to an inanimate object, image, or drawing: *anima* means soul in Latin. Animation is the art of movement expressed with images that are not taken directly from reality (Fig. 10.1.1). In animation, the illusion of movement is achieved by rapidly displaying many still images—or frames—in sequence. The stories behind animated images are critical to the success of a project, but the artistic use of the craft and techniques are also essential.

The first animated flipbooks and films were created at the turn of the nineteenth century, and the craft of animation grew during the first three decades of the twentieth century. But the classic principles of character animation were developed and perfected with the hand-drawn cartoon animations of the late 1930s and 1940s. At the time, animation borrowed many of its principles from older crafts and forms of popular and artistic expression including *vaudeville* shows, the visual arts, theatre, and cinema. Some of the computer animation techniques used to create **sequences of still images** are based on the techniques and principles of traditional character animation, others are based on the findings of experimental animators and filmmakers, and many are unique to the new expansive medium of computer animation. Many early animation techniques—such as hand-drawn, stop-motion, time-lapse, and performance animation—have been adapted to the new digital tools and are still used in computer character animation, effects animation, and computer-generated visual effects.

(Top: Detail of *For the Birds*. (© Disney Enterprises, Inc./Pixar Animation Studios.)





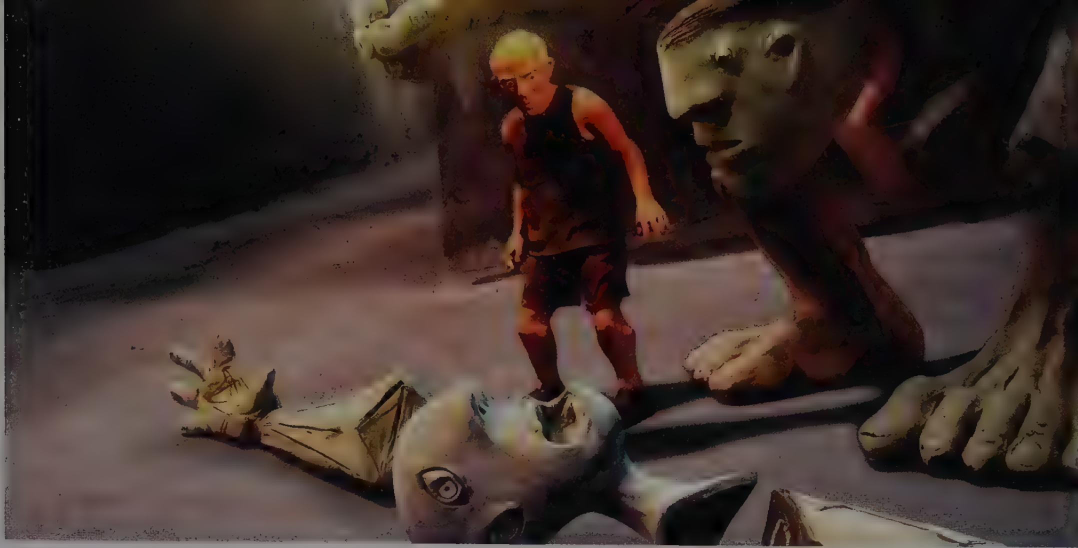
10.1.2 (Top left) Shooting flat artwork under a camera stand; painting one frame at at time (top right); animator at work using a light-table and computers. (© 2008 Royal College of Art. Photography by Passionpixels.)

Hand-Drawn Animation

The most common technique of traditional animation is **hand-drawn animation**, often called **cel animation**. This technique starts with sequences of individual drawings on paper (Fig. 10.1.2). These drawings are recorded successively on an animation stand to create a preview of the motion, or **pencil test**. Once this rough animation is approved, the drawings are then sent for **cleanup**. The cleaned-up drawings then go to **ink and paint**, which means that the pencil lines are inked and the areas of the drawing that require color are painted. Years ago the inking and painting of animation drawings was done entirely by hand on individual acetate overlays or **cels**, but the use of cels is on the decline. Today we scan the cleaned-up drawings into a digital ink and paint system and save them as files. The digital files with the scanned drawings contain the foreground shapes that move over the background. The **foreground** may contain, for example, drawings of cartoon characters, letterforms, or scanned photographs; the **background** usually consists of painted or photographic images. In any given shot there may be several layers of drawings arranged on the background. Once finished, all the layers are composited digitally over the background. In the olden days this was achieved by placing the transparent cel overlays over the background and recording them on photographic film one frame at a time.

Stop-Motion Animation

Traditional **stop-motion animation**, also known as stop-motion photography, consists of animating a jointed model and recording the different positions on a single frame each (Figs. 10.1.3 and 10.1.4). This technique was successfully used to create landmark visual effects for live action films from the 1930s to the 1950s, including *King Kong* and *Jason and the Argonauts*. Early miniature models for tradi-



tional stop-motion animation were made of modeling clay, but most later models consisted of a rubber skin with a wire armature. Stop motion is a form of forward kinematics animation, and can also be used to set key poses of three-dimensional computer-animated characters. This is usually done with special metal armatures that send information on the joint angles to the animation software (Fig. 11.2.2).

10.1.3 A stop-motion Pablo Picasso encounters a few of his characters in *Minotauromaquia*. (© IB Cinema. All rights reserved.)

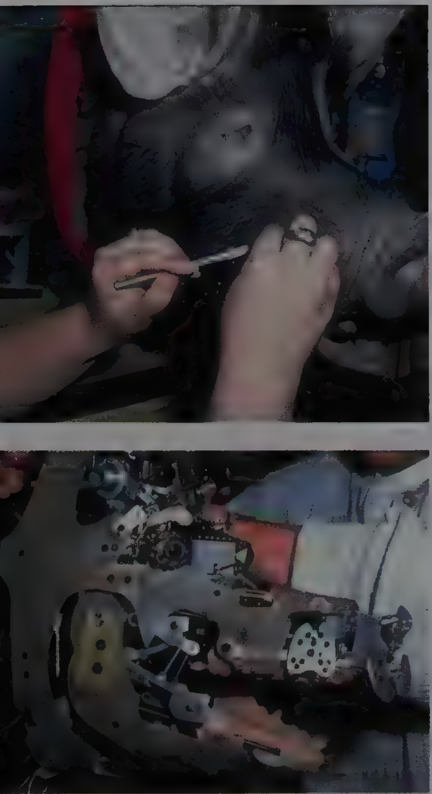
Animatronics

The computer-controlled models that can be animated in real time are called **animatronics**. These motion control systems have mechanical and electronic components, and they usually consist of a metal-jointed armature that is covered with a synthetic skin and moved with servomotors (Fig. 10.1.5). The fact that animatronics are usually placed on the set alongside live actors sometimes eliminates the need for compositing later in the production process. The motions of animatronics are usually programmed with forward and inverse kinematics software, and they may be repeated and refined since they are stored as digital information.

10.1.4 Metal armature used for the *The Werepig* stop-motion characters. (© IB Cinema. All rights reserved.)

Performance Animation

The oldest form of **performance animation** is probably found in the art of **puppetry**. *Bunraku*, for example, my personal favorite form of puppetry, exposes the puppeteer alongside the puppet. Whether created with puppets or with an actor inside of a suit, the basic idea is that a live actor controls the performance of the animated character (Figs. 10.1.6 and 10.1.7). When applied to three-dimensional computer animation this technique is usually called performance capture or motion capture, and it has two modalities: live motion capture, which is used to animate the computer character in real time, and processed motion capture, where the collected data is fine-tuned and



10.1.5 A servo-driven animatronic pig's head under construction (top), and the hair being punched into the synthetic skin that covers it. (Courtesy of Jim Henson's Creature Shop.)

enhanced with the use of other animation techniques (Figs. 12.2.3 and 12.2.7). When executed with skill, motion capture can bring to computer animation some of the freshness and natural motion of a live performance.

Character Animation

Character animation seeks to bring to life imagined or virtual characters, and is considered by many the highest form of animation (Figs. 10.1.1 and 10.4.1–10.4.17). In the majority of computer animation productions a few character animators are charged with blocking out the primary motion of the character, while assistant animators are responsible for cleaning up the primary motion and adding the secondary motion. **Character animation** is usually created with a combination of inverse and forward kinematic techniques and motion capture, both of which are covered in Chapters 11 and 12.

Effects Animation

Most of the animation that is not character-oriented falls within the specialty of **effects animation**. This usually includes natural phenomena like fire, smoke, wind, dust, and water in its many forms (rain, snow, clouds, rivers, waterfalls, oceans), as well as special lighting effects like sparks and shadows (Figs. 10.1.8 and 12.3.3–12.3.6). Effects animation often deals with props and entire sets, such as the ball in a soccer game or the grass in a landscape. This area of computer animation usually relies on techniques that allow the animator to control vast numbers of elements over time, such as particle systems and other procedural techniques and dynamic simulations.

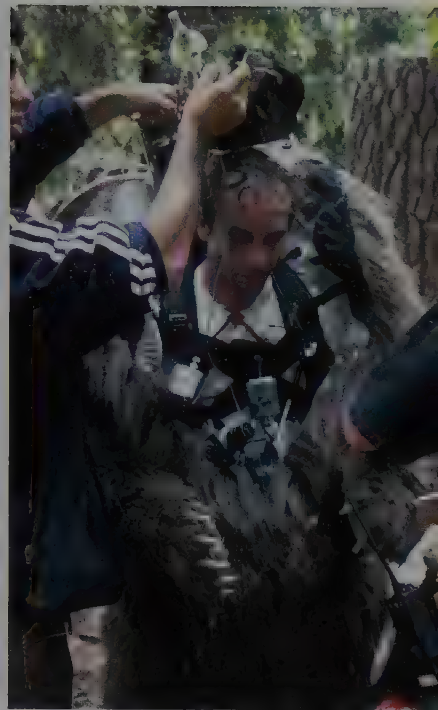
Visual Effects Animation for Live Action

Animating visual effects and characters for live action requires a unique approach that is usually quite different from traditional animation. Since the main goal of **visual effects animation** is to complement live action, most animated elements must visually match the motion, colors, lighting, and perspective of the live sequence (Fig. 11.2.6). Unlike cartoon animation where the creator is free—even required—to exaggerate motion, visual effects animation must blend seamlessly with the action plates provided by the cinematographer. (See Chapter 13 for more information on effects and character animation for live action.)

Keyframing and In-Betweening

One of the fundamental techniques used in animation is called **keyframing**. This technique is used to define an animated sequence based on its key moments. In the case of hand-drawn animation the drawings that correspond to the key moments in an animated

sequence are called keyframe drawings, **keyframes** or **extremes** (Figs. 9.8.3 and 11.1.1). In stop-motion animation the key moments captured in the armatures or clay models are called **key poses**. Another animation technique called **in-betweening** is used once the keyframes have been established and drawn. In-betweening consists of creating all the transition or **in-between** drawings that fill in the gaps between the keyframes. In traditional animation, the in-betweening process is done by laboriously creating each in-between drawing by hand. In computer animation, in-betweening is usually done with a technique called **interpolation**. A variety of computer interpolation techniques can be used to create as many in-between frames as needed by using simple information, such as keypoints in a keyframe or interpolation curves. (See Chapter 11 for more information on interpolation.)



10.1.6 A suit performer in an animatronic bear costume is given a rest break between shots. (Courtesy of Jim Henson's Creature Shop.)

Units of Animation

Animations are made of thousands of frames, but the smallest unit of animation is the **frame**. One frame consists of a single still image, and for that reason one frame of animation is sometimes called a **still frame**, or simply a still. The number of frames that constitute one second of animation depends on the output media on which the animation is delivered. One second of animation at normal-speed video equals **30 frames**; one second on film equals **24 frames**. On an interactive real-time computer the frame rate adapts itself to the hardware capabilities, ranging from 8 to 60 frames. The number of frames of animation per second is also called the **rate of display** or rate of projection, and is usually indicated with the letters **fps (frames per second)**.

Animation that is created by recording a different drawing on each frame recreates motion with the highest quality. One drawing (or image) per frame delivers the most information about motion to our perceptual system. When budgets or deadlines are tight, animation is often created by recording each drawing on two consecutive frames. This is known as **shooting on twos**, and the resulting quality may range from acceptable to almost as good as when each image is recorded on a single frame. Most animated series created for television and many feature animated films today are shot on twos. When animated within high standards the quality degradation goes unnoticed by most moviegoers.

10.1.7 Preparing to shoot the animatronic dog with a walking mechanism, mounted on a rolling rig along with the puppeteer, and towed behind the camera dolly. (Courtesy of Jim Henson's Creature Shop.)

In film and video production, a **shot** is a string of frames recorded by a single camera without interruption. A shot can consist of a few frames, or it can go on for several seconds or even minutes. A **sequence** consists of a succession of camera shots that are connected to each other because they develop the same aspect or moment of the action. Several sequences of shots usually add up to a scene. A **scene** can also be described in a more traditional way as continuous action in one place or as a unit of traditional storytelling. An **act** in a traditional theater play is usually made up of several scenes. In some animated productions a shot is sometimes referred to as a scene.

10.1.8 Effects animation includes animating lighting effects and the flight paths of props like the vehicles in this detail from *Lights & Water*. (Courtesy of Satoshi Kitahara.)

10.1.9 (Opposite page) This TV commercial exemplifies parallel action, which is at the core of this work, implemented as a montage of live action and computer-generated short sequences (leaving the beach, arriving at the mansion) and single shots. At the end of the sequence the image format changes from letterbox to full screen format. (Stills from: *Lara Needs SEAT*. Client: SEAT. Agency: Callegari-Berville. Production: Ex Machina. Director: Pascal Vuong. Consultant: Eidos Interactive. Images courtesy of Ex Machina.)

The style of the great majority of today's computer animations certainly relies more on the juxtaposition, or **montage**, of short shots than on a few long shots. The clarity of visual storytelling has to do as much with the composition of each shot as it has to do with the montage of shots and sequences.

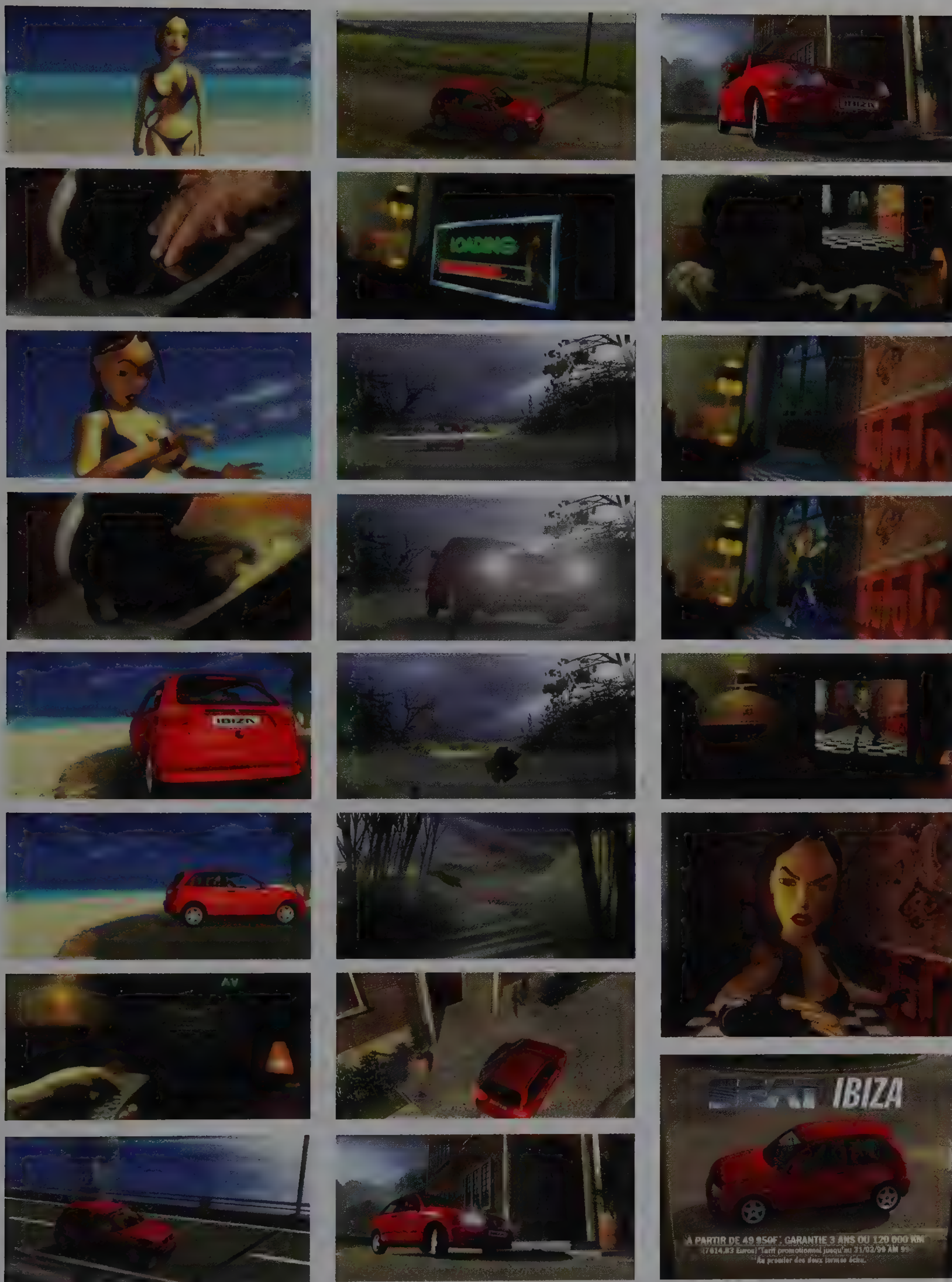
Timing of the Action

Telling a story with computer animation is based on timing the actions to the story and on designing motions that convey the desired effect. The **timing of the action** is based on how the actions of the characters and the motion of the camera are timed to the story. There are many ways for the visual action to relate to the story. The action may be ahead or behind the story, or it may be shown in parallel.

The action may be **ahead of the story** in cases when, for example, an animated character reacts to a sound located off-camera by turning its head. The character's action is ahead of the story because it indicates to us that something will happen before we know what it is. The action may be **behind the story** when the audience knows before the character what is going to happen next. The action is behind the story, for example, in a long shot where the audience can see that a piano is falling from the roof of a building right over a character who is unaware of the impending and disastrous action. *Knickknack*, a 1989 animation pictured in Figure 1.3.4, is a classic example of the action being constantly behind the story. Timing the action to be slightly ahead or behind the story is a good technique for keeping the interest of the audience. The effect of actions that are ahead of the story may be suspense and expectation because the audience wants to find out the outcome of the story and they try to guess what's next. The effects of the actions that are behind the story are commonly used in comedy, and audiences enjoy it because they can watch the character find out—often the hard way—what they already knew. **Interrupted action**, also called a **fake hold**, can be used to give the audience a moment to catch up with action that is ahead of the story, or to savor action that is behind the story.

Parallel action occurs when the audience is shown actions that take place at the same time but in different places. Parallel action is often shown as a series of shots that cut back-and-forth between each location (Fig. 10.1.9). Parallel action may show, for example, a man having lunch at a restaurant with his friend, while their respective wives are having lunch and talking about their husbands in a restaurant that is located around the corner from where their husbands are. Parallel action can be used to indicate a change of events or a turning point in the story. The action in an animation can also be interrupted to hold the interest of the audience for an unexpected action that is about to occur.

The skillful timing of the action in an animated project, or in any visual storytelling for that matter, can have a major positive effect on the audience—to hold their interest by having them con-


LOADING
IBIZA
IBIZA
A PARTIR DE 49 950F . GARANTIE 3 ANS OU 120 000 KM

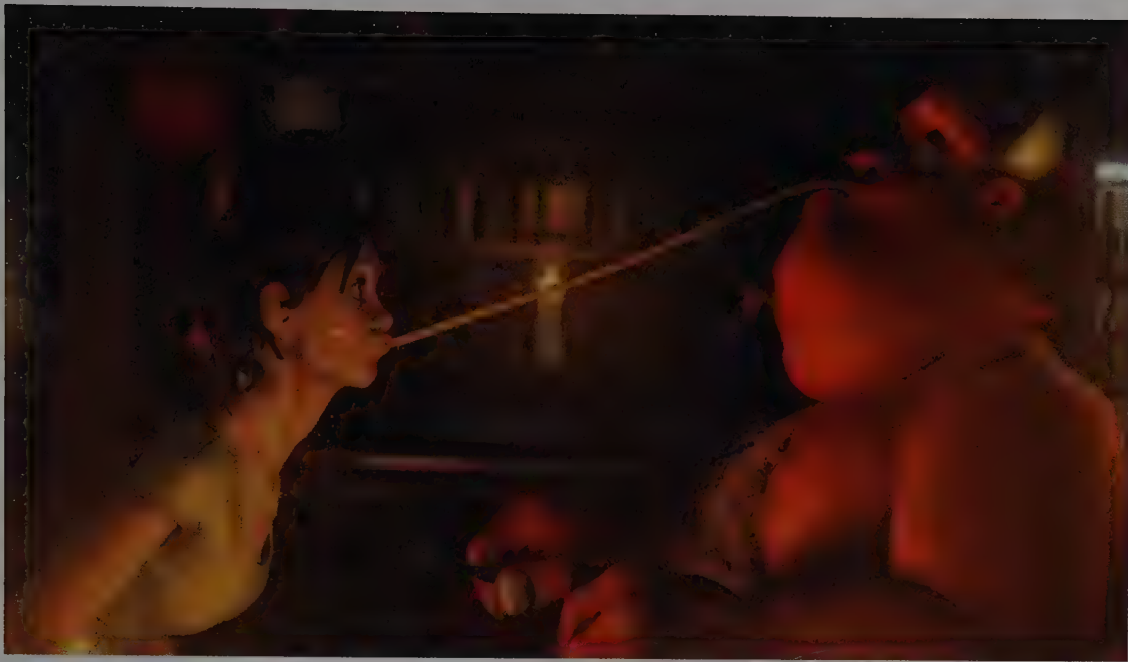
10.1.10 The surprise takes the form of a motion hold, where these *Dragon Hunters* characters pause for a second to think about and digest the event they are witnessing. (© MMVII Futurikon Films, Trixter, LuxAnimation, France3 Cinéma, RTL-Tvi, in coproduction with Mac Guff Ligne.)

stantly trying to guess how the story will evolve. This sense of curiosity and **anticipation** of the action is essential to all successful visual stories. Equally important to a good story is the **follow-through** of the action. Good follow-through keeps the interest of the audience by allowing them to confirm their expectations or suspicions or to surprise them with an unexpected turn of events. In any case, the visual follow-through of the action gives the audience a chance to digest and enjoy the plot of the story.

The Visual Grammar of Motion

The motions of a character help to tell the story and to define the personality and emotions of the animated character. (In cases when the computer animation is based on abstract shapes and not on characters, the motion of the shapes and the timing of the action becomes the main conduit for storytelling due to the lack of facial expressions and body gestures.) Motion is also a great tool for directing the **attention of the audience** to a specific place in the image. For example, a slight motion in the background of a calm scene will immediately draw the eyes of the audience to that area. Motion is so effective for guiding the eyes of the audience that it should be carefully choreographed. The **readability of motion** will result in action that flows, while confusing motion will result in unfocused action.

Different combinations of timing, speed, rhythm, and choreography result in different types of motion, including primary and secondary motions, overlapping motions, staggered motions, and



10.1.11 The moment of closeness for the protagonists of *My Date from Hell* is approaching, and that story is told visually by linking the two with a piece of spaghetti. (© 2006 Filmakademie Baden-Württemberg, Tim Weimann, Tom Bracht, and Patrick Wachowiak.)

motion holds. These types of motions apply not only to the objects and characters in the environment but to the camera. The virtual camera plays an important role in computer animation because its motions—as well as its position, point of interest, and focal length—have a powerful storytelling effect (see Chapter 7 for more information on camera animation).

To choreograph motion is to compose and arrange the motion of all objects or the parts of objects in a sequence of actions. Choreographing motion starts with planning the action and breaking it down into manageable blocks. This becomes very useful when animating characters with many parts that move at the same time and is also helpful to refine simple movements. **Simple motion** may consist of just one object or part of the object moving in a single direction, while **complex motion** may consist of several objects or their parts moving in a variety of directions, speeds, and rhythms. In computer animation, most environments with multiple models, and most models—particularly characters—with multiple joints imply complex animation.

Complex motion consists of primary or dominant motions and secondary motions. The **primary motion** in a shot is the motion that captures the audience's attention. The primary motion in a character is the motion that carries the action forward. The primary motion in an interior shot, for example, could be personified by two loud patrons in a bar who laugh hysterically while the rest of the customers watch. The **secondary motion** in a shot is the motion that echoes or complements the primary motion. The arms slowly

Traditional Principles of Animation

1. Squash and stretch
2. Anticipation
3. Staging
4. Straight-ahead action and pose-to-pose
5. Follow-through and overlapping action
6. Slow-in and slow-out
7. Arcs
8. Secondary action
9. Timing
10. Exaggeration
11. Solid drawing
12. Character appeal

10.2.1 The principles of animation developed by cartoon animators in the 1930s. These principles are about acting, directing, representing reality, simulating physics, and editing motion.

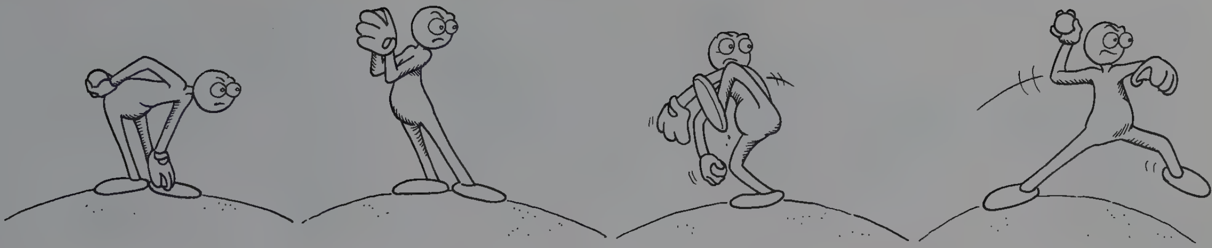
(Pocoyó surrounded by friends. Pocoyo™ and Zinkia™ © 2005 Zinkia Entertainment SL.)

moving closer to the body of a spinning ice skater or the snapping fingers of a jumping dancer are both samples of secondary motion. Secondary motion often starts as a reaction to a primary motion and through time becomes the new primary motion.

Motions in a sequence—whether just primary motions or a combination of primary and secondary motions—are rarely independent from one another. Motions often alternate and overlap. **Overlapping motion**, also called staggered motion, occurs when some motions start before others conclude (Fig. 10.2.7). Our world is full of overlapping motion. For example, when people walk they start moving the left foot before they have finished moving the right foot. When a tree is moved by the wind the small branches move at one rhythm, the larger branches at a different rhythm, and the leaves move at yet a different rhythm. For this reason, animations that want to replicate or echo natural motion are usually based on the principle of overlapping motion: before one motion dies a new one starts to bloom.

A **motion hold** happens when a character interrupts or concludes one motion and pauses. The function of a motion hold is to give the audience a chance to catch up with the development of the story or to indicate that a new action is about to happen (Figs. 10.1.10, 10.1.11, 10.4.10, and page v). The most effective way to create a motion hold is by interrupting the primary motion while continuing with a small motion. Motion holds should never result in an absolute and total interruption of the action because that results in mechanical motion or destroys the continuity and flow of the action. Secondary motions such as the turning of a head or blinking of the eyes can help carry on a motion hold.

The relation between primary, secondary, and overlapping motions and motion holds becomes an important concern when animating a group of moving objects, especially when animating articulated figures with hierarchical groupings of objects. A useful strategy for planning and refining complex motion in computer animation is based on the idea of animating the **layers of motion** one at a time. An animated scene can have several layers of motion. In a scene with primary, secondary, and overlapping motion it usually makes sense to animate first the layer of primary motion, then add layers of secondary motion, and finally go back and forth between layers to adjust all the overlapping motions. Animated objects or characters can also have several layers of motion, and the most practical way of animating them consists of starting with the primary motion and doing the secondary motions next. When animating an articulated figure with keyframes, this implies establishing the poses for each keyframe starting at the top of the hierarchical structure and working down to the details only after the dominant motions have been worked out. In a scene that consists, for example, of one singer and a five-character chorus it would be convenient to start animating the primary motion of the lead singer. One could continue with the dominant motions of the chorus singers. Then the secondary motions of the lead singer could be determined, followed by the



secondary motions of the chorus singers. At this point fine-tuning the overlapping motions would be easy because all the primary and secondary motions would already be contained in the scene.

Animating motion in layers is convenient because it breaks a complex challenge into smaller and more manageable parts. It is also convenient and almost necessary in complex animated sequences because in many of them the keyframes in different motion layers are placed in different moments along the timeline. Keyframes in an animated sequence with complex motion are usually abundant and scattered throughout the scene in an overlapping fashion.

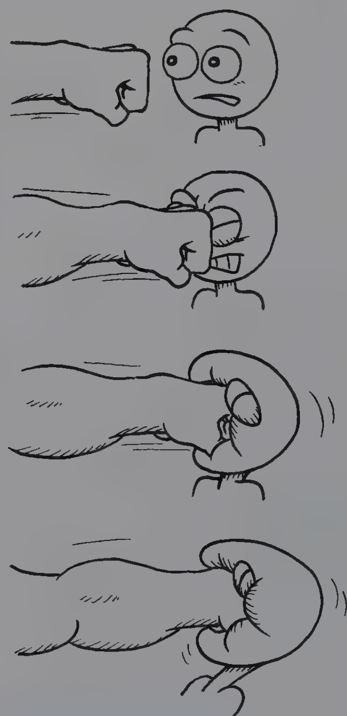
ANTICIPATION

10.2.3 Before throwing the ball the pitcher anticipates the target and winds up for the pitch.

10.2 The Twelve Principles

The **twelve principles of animation** were created in the early 1930s by animators at the Walt Disney Studios. These principles were used to guide production and creative discussions as well to train young animators better and faster (Fig. 10.2.1). These twelve principles became a foundation of hand-drawn **cartoon character animation**, and helped to transform animation from a novelty into an art form. By applying these principles to their work these pioneering animators produced many of the earliest animated feature films that became classics: ***Snow White*** (1937), ***Pinocchio*** and ***Fantasia*** (1940), ***Dumbo*** (1941), and ***Bambi*** (1942). The original principles are still relevant today because they help us to create more believable characters and situations. They can be applied to any style of animation, even though many of them work best for a cartoon comedy style. A few new principles to address the new techniques and styles of three-dimensional computer animation are suggested later in the chapter.

Early animation borrowed from previous crafts and forms of expression including cinema, theatre and the visual arts. This is evident when we realize that the twelve principles are mostly about five things: telling a story, acting and/or directing the performance, representing reality, the craft of representing or creating a reality in a believable way, and editing a sequence of actions.



SQUASH AND STRETCH

10.2.2 The first of the twelve principles developed by Disney animators in the 1930s, squash and stretch is used here to distort a face after receiving a punch.

Principle 1: Squash and Stretch

Squash and stretch, the first principle of the twelve, is used to exaggerate the amount of non-rigid body deformations usually with the purpose of achieving a comedic and/or dynamic effect (Fig. 10.2.2).

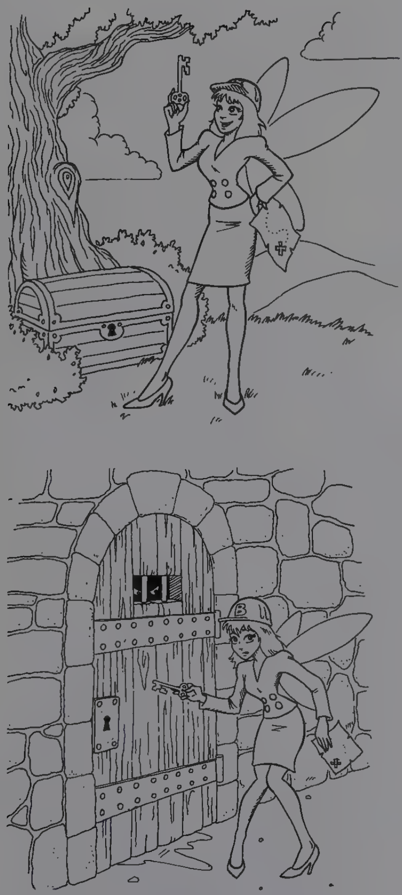
STRAIGHT-AHEAD ACTION

10.2.4 The improvisation, spontaneity, and unpredictability in the walk of a drunken character illustrates straight-ahead action.

Typically bodies are squashed when they compress or stop, and they are stretched when they expand or accelerate. Three-dimensional squash and stretch can be implemented with a variety of techniques: skin and muscle, springs, direct mesh manipulation and morphing. It can also be implemented in more experimental ways with weighting, especially for dynamics simulations, and special-purpose rigging systems (Fig. 10.4.10).

Principle 2: Anticipation

The storytelling and acting technique of anticipation helps the audience predict what the next move of a character might be, or what might happen next in a story (Fig. 10.2.3). **Anticipation** is related to how long the character thinks, or not, before doing something. Anticipation deals with the fine balance that exists between keeping the audience behind the story and ahead of the story. The less we anticipate in animation the more surprising the action will be; the more we anticipate the more we "announce the surprise" that is about to happen in a scene. More anticipation (ahead of the story) usually, but not always, results less suspense (behind the story). Horror films, for example, take us on emotional rollercoasters by switching back and forth between lots of anticipation to total surprise. Motion holds, explained earlier in the chapter, can also be used as a form of anticipation. In three-dimensional computer animation anticipation can be fine-tuned using digital time-editing tools such as time sheets, timelines, and curves.

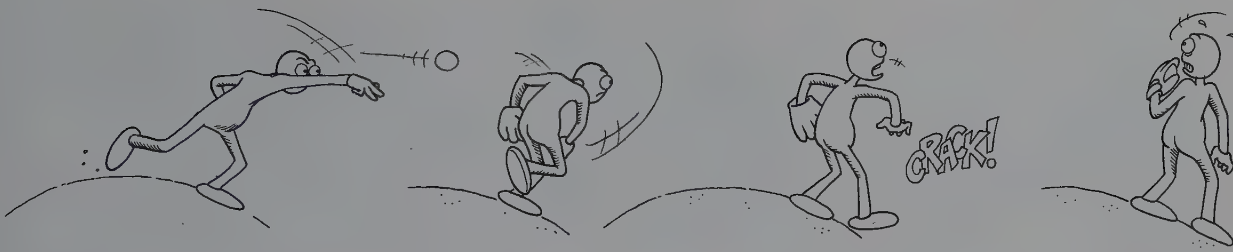


STAGING

10.2.5 Differences in staging can turn the same scene from happy to suspenseful.

Principle 3: Staging

Staging, or *mise-en-scène* as it is also known, is about translating the mood and intention of a scene into specific character positions and actions. Staging the key character poses in the scene is key to defining the nature of the action (Fig. 10.2.5). Two effective staging techniques to tell the story visually include hiding or revealing the center of interest (first story beat in Fig. 10.2.12), and using a chain reaction of actions-reactions (about to happen in Figure 10.2.7). Staging can also be aided with contemporary cinematic techniques such as slow motion, fast-forward and looped motion, frozen time, and hand-held camera moves. Three-dimensional animatics are a



FOLLOW-THROUGH

10.2.6 Follow-through includes the reactions of a character after an action. These let the viewer know how the character feels about what has just happened.

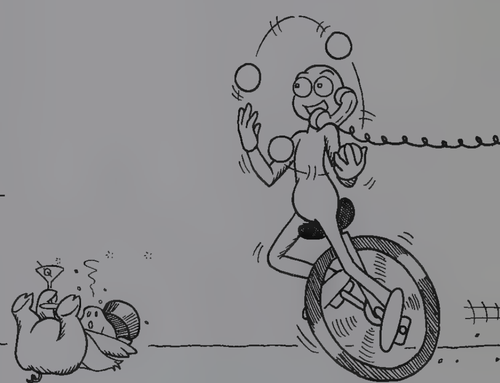
great tool for blocking out the staging of a scene before the primary, secondary, and facial animation are created (Figs. 2.2.6 and 2.7.10).

Principle 4: Pose-to-Pose and Straight-Ahead Action

Straight-ahead action and pose-to-pose are two different methods for animating a character, and they yield different results. **Pose-to-pose action** became the standard animation technique during the classic days of hand-drawn animation because it breaks down structured motion into a series of clearly defined key poses (Fig. 10.2.9). These key poses are captured in the keyframes. In **straight-ahead action** the character moves spontaneously, sometimes erratically, through the action one step at a time until the action is finished (Fig. 10.2.4). Motion capture, dynamics simulations, and three-dimensional roto-scoping, can easily be used as straight-ahead animation techniques. Straight-ahead motion can be blended with pose-to-pose motion through the use of animation channels.

Principle 5: Follow-Through and Overlapping Action

Follow-through and overlapping action are two techniques that help make the action richer with detail and subtlety. **Follow-through action** consists of the motions and actions derived from an initial previous action of the character. Follow-through adds physicality and detail to an action, and it can also be used to express how the character feels about what he/she just did (Fig. 10.2.6). **Overlapping action** is usually the compound result of multiple motions that blend and influence the position of the character. In three-dimensional computer animation a lot of the common follow-through motions of clothing and hair, for example, can be animated with dynamics simulations (Fig. 10.2.7). The layers and channels in three-dimensional computer animation software allow us to mix and blend different overlapping motions from different areas of the character.

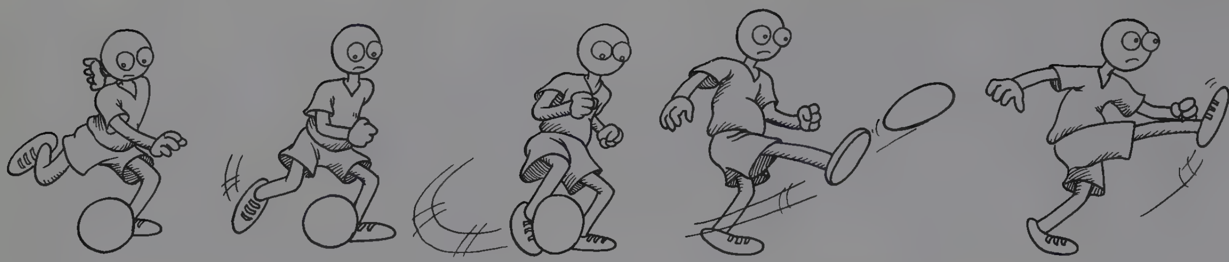


OVERLAPPING ACTION

10.2.7 Multiple motions happening at the same time result in overlapping action, for example when a character falls.

Principle 6: Slow-In and Slow-Out

Slow-in and **slow-out** consist of slowing down the beginning and the end of an action. A snappy effect is achieved when motion is delayed in this way (Fig. 10.2.8). These techniques may be com-

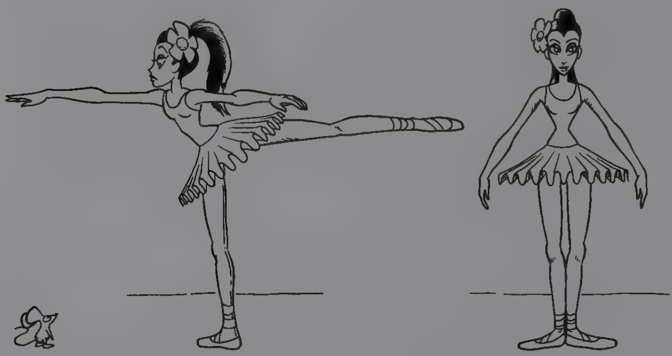


SLOW-IN...

...AND SLOW-OUT

10.2.8 Right before and right after kicking the ball a soccer player slows-in and slows-out; this highlights the precise moment of the kick.

bined with other animation principles: slow-ins can be used to anticipate while slow-outs can be used to follow-through. In three-dimensional computer animation slow-ins and slow-outs can be fine-tuned with digital time-editing tools. When using motion capture techniques for cartoon-style animated characters, it is convenient to remind performers to do slow-ins and slow-outs. Not mentioned in the original principles but a useful addition is the inverse variation of slow-ins and outs: a **fast-in** and a **fast-out**. This effect is often seen in TV commercials and music videos where the beginning and



ARCS, AND POSE-TO-POSE

10.2.9 The smooth motions of a ballerina follow curved paths. Her structured movements, from one key pose to another, are a good example of pose-to-pose action.

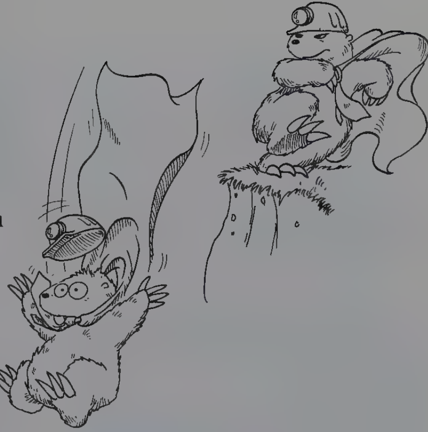
end of the sequence are accelerated while the middle is slowed down, giving it a surreal or dreamy feeling. An unusual effect of acceleration and deacceleration can also be achieved by speeding up the middle of an action.

Principle 7: Arc Motion

Using **arcs** to animate the movements of characters helps achieve a natural look because most living creatures move in curved paths, rarely in perfectly straight lines (Fig. 10.2.9). When seen next to arc motion, a non-arc motion comes across as sinister, restricted, or robotic. In three-dimensional computer animation we can use software constraints to force all or some of the motion within arcs. Motion-captured performances can also be fine-tuned with curve editors, as long as the motion is not flattened.

Principle 8: Secondary Action

Secondary action consists of the smaller motions that complement the dominant action (Fig. 10.2.10). For example, raising an eyebrow while turning the head, opening the hand while moving the arm to reach for something In three-dimensional computer animation we can take advantage of layers and channels for building up different secondary motions, for example, a layer for hair, a layer for the character's hat, a layer for the cape, and so on.

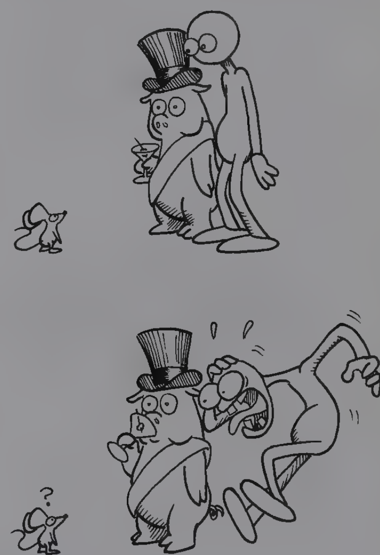


Principle 9: Timing

Timing is both the precise moment at which the character moves or acts, and the amount of time that he/she spends on the action. Timing is all about adding or subtracting emotion and intention to the character's performance (Fig. 10.2.11). A character in front of a mirror who is about to pick up a comb, for example, can extend his hand slowly to reach for it while looking at him/herself in the mirror or grab it in a light and casual way. The same character could comb his hair slowly while examining the result in the mirror, or comb quickly and distracted while looking around for something. The timing of actions in an interaction between characters can also be used to say a lot about the emotional state of the character: waiting, interrupting, pausing, thinking are all actions related to timing. Most three-dimensional computer animation tools allow us to fine-tune the timing by shaving off or adding frames with non-linear time editing. Timing can also be controlled and adjusted by placing each character on a separate track, and using sub-tracks for parts of the character such as head, torso, arms, and legs.

SECONDARY ACTION

10.2.10 Secondary action includes the motion of hair, fur, clothing, and accessories.



TIMING

10.2.11 The timing of a response to someone calling us, or to a noise, tells a lot about our attitude, confidence, and disposition.

Principle 10: Exaggeration

Exaggeration is about taking motions and actions to extreme states, either magnifying a quality or taking one away. Happiness, for example, could be serene and intimate but an exaggerated happiness could be manic and devoid of equilibrium. Exaggeration helps cartoon characters deliver the essence of an action (Fig. 10.2.12). Cartoon characters themselves are caricatures, or representations with exaggerated features and manners. Exaggeration can be complemented with a few other principles—in particular squash and stretch, slow-ins and outs, and follow-through. In three-dimensional computer animation we can use procedural techniques, motion ranges, and scripts to exaggerate motion. Traditionally this principle referred to the performance of the character. But exaggeration can also be applied to the cinematography and editing, not just the performance, to increase the intensity of a moment.

EXAGGERATION

10.2.12 Exaggerated reactions are often comedic.

SOLID DRAWING

10.2.13 Solid drawing helps to pose silhouettes that are easy to read visually, and good rigging helps animators deliver performances that are adequate to the character. The equivalent of this principle in 3D computer animation is solid modeling and rigging.

Principle 11: Solid Drawing

The principle of **solid drawing** is about the clear delineation of shape that is necessary to bring animated characters to life (Fig. 10.2.13). In hand-drawn animation clear delineation is accomplished through drawing. In three-dimensional computer animation, however, clear delineation is accomplished through modeling, rigging, and even lighting. **Clear delineation** would be a more encompassing name for this principle. Clear delineation in image-making is like good diction in speech. Good three-dimensional modeling, with the help of lighting and good cinematography, helps to convey the weight, depth, and balance of a character. The clear silhouettes of characters as seen by the camera help delineate the pose and intention. Animation rigs that are optimized for a specific type of motion help reveal the personality of the character.

CHARACTER APPEAL

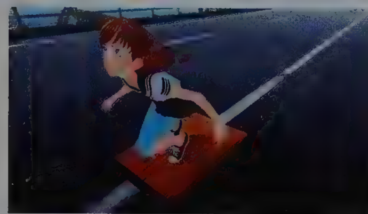
10.2.14 Understanding the motivations and desires of characters makes their personality better defined and increases their appeal.

Principle 12: Character Appeal

Character appeal is about how well-rounded, believable, and unique a character is. Character appeal facilitates the emotional connection between character and audience (Fig. 10.2.14). Appealing characters are well developed, have an interesting personality, and have a clear set of desires or needs that drive their behavior and actions. Attractive or repulsive, characters can be appealing as long as their performance impacts us. Complexity and consistency of motion are two elements of character appeal that can be easily developed with three-dimensional computer animation. Writing down the ways in which the character moves (Fig. 10.4.7), how he/she reacts to different situations, and how he/she relates to other characters can help define the main characteristics of the character's personality. Fine-tune the personality with key poses and character turnarounds.

10.3 Half-a-Dozen Additional Principles

The traditional twelve principles of animation were put together in the 1930s, but animation techniques and styles have changed tremendously since then. The scope of productions and the world around animation have also changed. The dominant, almost exclusive, style of animation in the 1930s was hand-drawn pose-to-pose cartoon narra-



10.3.1 Limited animation in the anime style with insinuated motions accentuated with visual swishes. (© Keica, Inc.)

tive animation. Today many styles of animation coexist in animated feature movies, independent shorts, as well as non-linear interactive games and non-narrative music videos.

In the 1930s many animation techniques and capabilities were underdeveloped, even misunderstood. Camera moves and lighting techniques—for example—were limited, and rotoscoping and stop-motion were considered lesser accomplished forms of animation than hand-drawing. Consider the new tools that have transformed our craft since the 1930s: computer animation, hand-held cameras, television, non-linear editing, compositing, motion capture, inverse kinematics, and procedural tools.

All artforms, animation included, have greatly evolved since the 1930s, creating new languages and new principles. For this reason I say that it is time to reinterpret and expand the original principles. We also need to add a few new principles that address today's new animation styles, technique, and possibilities. This is our collective challenge. I propose a few new principles for computer animation including: limited animation, cinematography, facial animation, visual styling, motion blending, and user-controlled animation (Fig. 10.3.2).

Half-a-Dozen New Principles for 3D Computer Animation

13. Limited animation
14. Cinematography
15. Facial animation
16. Visual styling
17. Motion blending
18. User-controlled animation

10.3.2 These principles contain additional tips and guidelines to produce better animation with today's new animation styles and technique.

New Principle (13): Limited Animation

Initially developed as a clever creative style to address restricted resources, **limited animation** was for a long time considered a lesser style incapable of delivering fluent motion. But throughout the decades limited animation developed into an expressive form with an extensive formal repertoire. There are many variants of limited animation but two have gained the most notoriety: Japanese anime (アニメ), and American TV animated series in the style tradition of 1960s Hanna-Barbera's work. **Anime**, also known as animé, encompasses today a great variety of styles based on the early work of animation pioneer Osamu Tezuka, creator of *Astro Boy* and *Kimba the White Lion*.

Generally speaking, anime renders human figures and faces in a

stylized way, including the extreme caricatures with large eyes so typical of the genre. Some of the limited animation trademark techniques used in anime include: abundance of motion holds; embellishment of static scenes with wind effects; minimal animation of facial expressions; extreme exaggeration of facial expressions on static bodies; camera POVs with extreme perspectives; looped character animation cycles over looped background motion; overlayed time-lapse motion; split-screen simultaneous parallel action; representation of motion and speed with swishing trail lines; and symbolic expression of emotional states through exaggerated sweating, blushing and trembling. Many of these techniques can be easily adapted to computer animation techniques (Fig. 10.3.1). Some TV animation and independent animation also take advantage of the expressive powers of limited animation. The example illustrated in Figure 10.3.3 shows a four-second limited animation sequence of a comedic dog walk that was put together with four drawings, each a different position: dog down with eyes open or closed, and dog up left or right. An additional one-second-long motion hold of a stunned pose concludes the sequence. The effect is hilarious.

10.3.3a Five drawings used to animate a *Guard Dog* four second walking sequence in a limited animation style. The frame sequence is in Fig. 10.3.3b. (Images courtesy of Bill Plympton.)

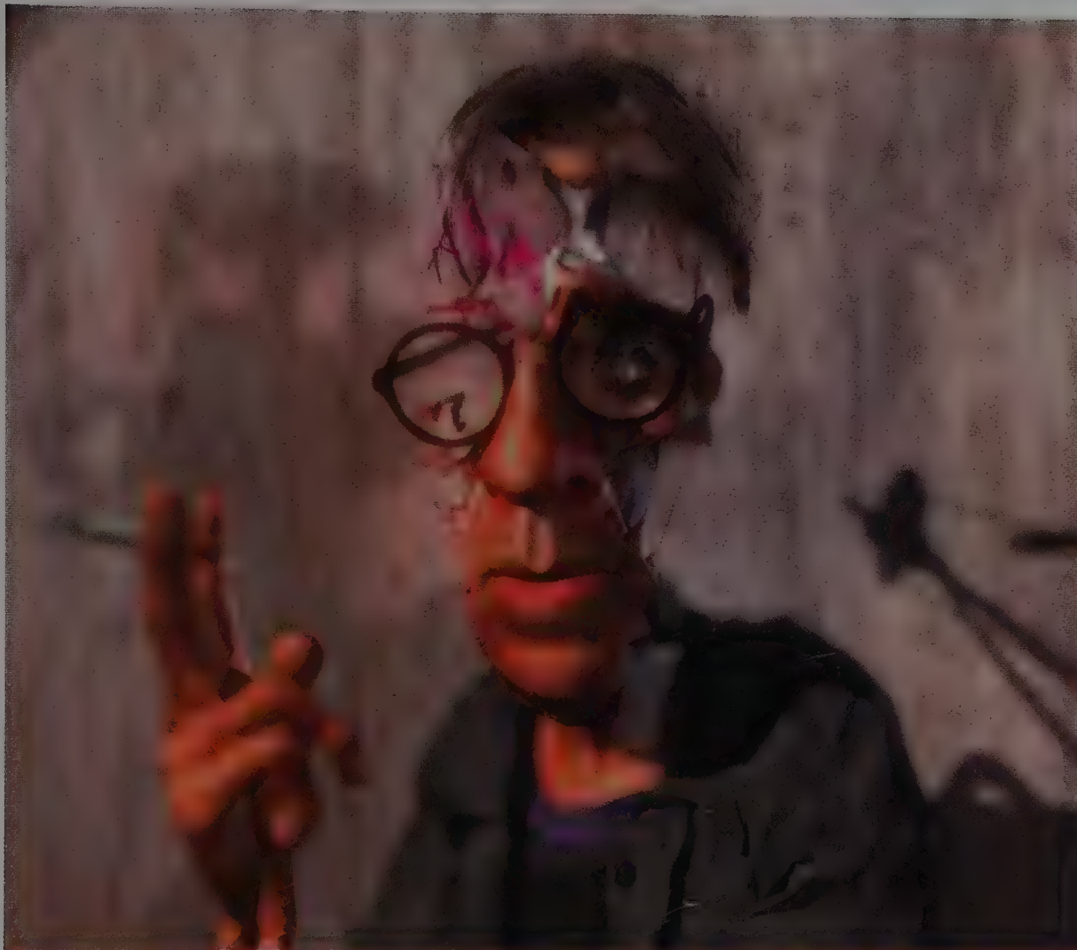
New Principle (14): Cinematography and Editing

Computer animation offers an unparalleled control of the camera and the lighting in a scene, and non-linear editing offers ease and flexibility in structuring an image sequence. These developments allow us to use the visual language of cinematography and editing to do a fair amount of storytelling. Gone are the days of early cinema and animation when a camera on a still tripod recorded the performance of actors on a stage. Likewise, editing is beyond the days where it was just about splicing together a predetermined order of shots. **Cinematography** and **editing** are crucial components of computer animation, not just afterthoughts or slaves to the performance. They can both be used to actively tell the story and not just to record the performances of the characters (Fig. 10.3.5).

New Principle (15): Facial Animation

The thoughts and emotions of characters can be expressed through their facial expressions, which are a privileged form of nonverbal communication and a quintessential human characteristic (Figs. 10.3.4 and 10.3.5). Three-dimensional computer animation offers more **facial animation** controls than any other animation technique, including very subtle motions of eyeballs, eyelids and small muscles. This technical capability can be enhanced by developing a consistent vocabulary of facial expressions, usually before production starts so that there is time to refine it and code it properly (Figs. 12.5.9 and 12.5.15).

The **Facial Action Coding System**, or **FACS**, developed by psychologist Paul Ekman and others is a useful resource, among others, in the creation of facial animation. The expressions of the Gollum

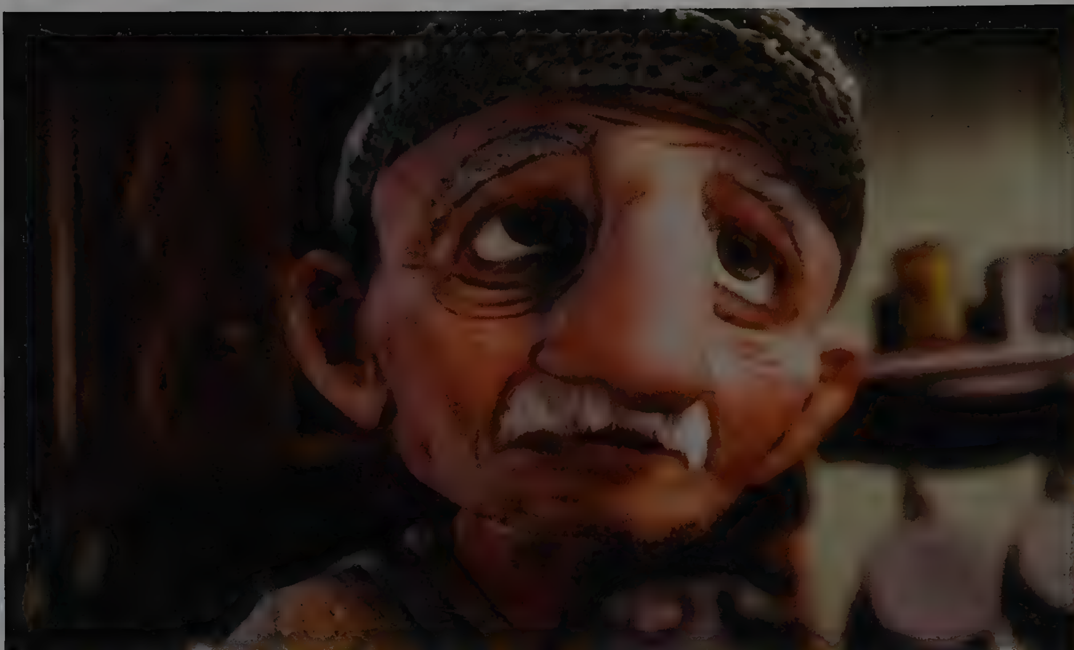


character in ***The Lord of the Rings: The Two Towers***, for example, were coded using FACS as a reference (Fig. 12.2.7). FACS highlights seven basic universal emotions, each with a family of facial expressions: anger, fear, sadness, disgust, happiness, surprise, and contempt (Fig. 10.3.6). These facial expressions of a character can be calibrated with these basic emotions, keeping in mind that there is a correlation between the intensity of expressions and feelings. The animation of these emotions can be further refined by drafting emotional profiles of the character, one of them based on the strength and the speed of his/her emotional response: moderate or strong, slow or fast—it's all about the timing! Another interesting FACS concept is the existence of rapid intense expressions, called microexpressions, that happen in a fraction of a second—typically 1/25th according to Ekman's research. Microexpressions are often related to concealed emotions.

Establishing early in the process the level of facial control and techniques has a positive effect on the styling of the character and

10.3.4 The story of *Ryan*, directed by Chris Landreth, is told with a creative mix of animation styles and narrative techniques. © 2004 Copper Heart Cut Inc. and the NFBC. All rights reserved. Used with permission of the National Film Board of Canada.)

10.3.3b *Guard Dog* sequencing of five drawings: Down Eyes Open (DEO) 3 frames, Up Left (UL) 4, Down Eyes Closed (DEC) 2, Up Right (UR) 6, DEO 2, UL 6, DEC 2, UR 6; DEO 2, UL 6, DEC 2, UR 6, DEO 2, UL 6, DEC 2, Up Right 6, DEO 2, UL 6, DEC 2, UL 6, DEO 2, UL 6, DEC 2, Stunned 30; for a total of 119 frames, about 5 seconds.



10.3.5 The expression of emotion and thought by the protagonist in *Même les pigeons vont au paradis* is focused in the eyes and the brow. Notice the large eyes, a facial feature usually associated with babies. (© BUF, Director: Samuel Torneux.)

the design of the production flow. It is useful to be familiar with the main muscles that drive facial expression. Depending on the level of character complexity we create a simple or a complex solution to drive facial expression, but always taking into account the emotions required by the character's situation and performance needs. Building an expressive repertoire or inventory of facial expressions is as essential as building walk cycles, whether we use the techniques of muscle systems, bones, motion capture, morph targets, or blend shapes.

List of Emotions in the Facial Action Coding System

1. Anger
2. Fear
3. Sadness
4. Disgust
5. Happiness
6. Surprise
7. Contempt

10.3.6 The Facial Action Coding System, or FACS, list seven universal emotions that are shown and understood through facial expressions.

New Principle (16): Visual Styling

Visual styling in three-dimensional computer animation means more than just rendering in a realistic style. The newer technologies allow us to explore a myriad artistic styles that make animation today one of the most innovative and *avant-garde* creative disciplines (Figs. 10.3.4–10.3.7). Visual styling is set during the visual development stage, and it can be used a storytelling device, for example, when we reinterpret the visual trademarks of a particular culture, artistic movement, or historical period (Figs. 2.3.1–2.3.5). Visual styling also adds expressive power to the story when it is fresh and appropriate. In other words, long gone are the days where good animation has to look like a Disney-style 1940s cartoon.

As we develop a visual look we must keep in mind what it is feasible to produce within the boundaries of the project, because certain looks might have a significant impact on rendering, on animation techniques, and on overall production complexity. A specific

style for rendering and animating hair or fur, for example, might look cool but might also require complex rigs, detailed models, and a complex animation process.

New Principle (17): Blend Motion

Good computer animation must have consistency of motion, but it can also be enriched by combining different styles of motion. It is possible today to easily **blend motion** from different sources, and combine multiple animation techniques on a single motion. Figures 10.3.4 and 10.3.5, for example, illustrate a project that successfully blends cartoon with realistic motion. Figure 12.2.3 illustrates a blend of stylized animation and performance capture. Before production starts it is necessary to define clear guidelines for whatever motion and animation styles will be utilized, including cartoon physics, realistic cartoon, realistic human motion, rotoscoping, and performance capture. The latter technique requires that we direct the live performers when capturing their motion to add intention to their movements.

There are many computer animation techniques for controlling the motion of three-dimensional characters. Some of these **motion control techniques**—inverse kinematics, for example—work naturally within the context of keyframe animation; others, including motion dynamics, require procedural animation methodologies that are borrowed from scientific simulations. Computer animations are developed today within a **hybrid framework** that combines different computer animation techniques and production methodologies

10.3.7 Using three-dimensional character techniques to animate this character gives the scene a surreal flavor. See Figures 6.9.3 and 9.2.1 for related images. (© Prima Linea Productions, www.FearsOfTheDark-themovie.com.)

10.3.8 The ability to decide what the protagonist does and where the camera looks gives audiences, gamers in particular, an unprecedented amount of creative control in new forms of entertainment. This creates a challenge for animators and game designers. (*Tom Clancy's Ghost Recon Advanced Warfighter*. © 2007 Ubisoft Entertainment. All rights reserved. Ghost Recon, Ghost Recon Advanced Warfighter, the Soldier Icon, Ubisoft, Ubi.com and the Ubisoft logo are trademarks of Ubisoft Entertainment in the U.S. and/or other countries.)

10.3.9 An early example of successful motion capture of two opponents. Their external shapes are realistic, and the internal structure that governs their motion is complex. Their facial expressions are limited in range but intense: pain and fear, determination and coldness. (Courtesy of Acclaim Entertainment, Inc., Advanced Technologies Group.)

into a single project. An important task in planning a computer animation is selecting the most appropriate technique or set of techniques for **implementing the motions** required in a specific project. The most common computer animation techniques are covered in Chapters 11 and 12.

Kinematic techniques for animating objects and characters are based on changing the position and orientation of models in three-dimensional space. In **forward kinematics,** the angles of the joints are manipulated to achieve a specific motion, while in **inverse kinematics** the limbs or objects are positioned and the software calculates the joint rotations that are necessary for creating the in-between positions. Inverse kinematics calculates the motion of entire skeletons by specifying the final angle positions of just some of the key joints that define the motion, and are useful for animating complex models with a large number of joints. **Motion capture** techniques provide the kinematic information to the software by recording the positions or angles of joints of live actors or objects in motion (Fig. 10.3.9).

Animation techniques based on the physical laws of motion, called dynamics, can generate realistic motion of objects by simulating their physical properties and nature's laws of physical motion. **Motion dynamics** techniques control the motion of three-dimensional objects by applying forces to the joints and actually simulating the motion that would result in the physical world if such forces

were applied to a real object with specific characteristics. Motion dynamics techniques take into account variables such as an object's weight, mass, inertia, flexibility, and collision with other objects, as well as the environment's friction, gravity, and other forces that may influence the motion of objects. **Procedural** or **rule-based motion** techniques animate the objects in the scene based on a set of procedures and rules to control motion. The animation of flocks is an example of procedural motion control. Rule-based motion techniques are often scored with a special-purpose programming language so that they can be easily edited and previewed.

10.4.1 Shrek's facial expression and paused hand gesture indicate that he is thinking about what he is about to say. (*Shrek*™ and © 2001 DreamWorks L.L.C.)

New Principle (18): User-Controlled Animation

Unlike a computer animated movie where every action has been planned in detail and carefully built within a chain of events, the control of the final actions and animations in computer and platform games is in the hands of gamers. A game offers multiple storylines that are chose as the game is played, and this poses the challenge to create great animation that works regardless of what move the gamer decides to make. Games are a combination of **user-controlled animation** and preset/narrative animation. One of the creative challenges in computer animation is to find a balance between the narrative and the improvisational aspect of games, a balance between the

fixed and the user-controlled. The combination of preset and dynamic user-controlled cameras is unique to games (Fig. 10.3.8). User-controlled animation relies on strong animation cycles with built-in anticipation that are able to branch smoothly into reaction shots—many of today's game engines can smooth transitions between animation cycles. Think, for example, of the nicely animated cycles of classic game cartoons like Nintendo's Mario or SEGA's Sonic the Hedgehog. They combined realistic moves with stylized animation, and even their idle resting positions had built-in personality.

10.4.2 Turnarounds present the character's key features and proportions. *Angelina Ballerina* was modeled with polygons NURBS was used to model the eyes. The high resolution full-body model has 47,439 polygons (facing page bottom). (© 2009 Helen Craig Ltd. and Katharine Holabird. By permission of HIT Entertainment Ltd.)

10.4 Character Development

Much of the story in an animated film is communicated through the actions of characters. For that reason a great deal of time and energy is dedicated to the character development stage in a project. In developing the look of a character for computer animation one must keep in mind the type of project that the character is intended for. Throughout this book you will find examples of cartoon characters, more stylized characters or very realistic. **Cartoon characters** are usually caricatures, representations with exaggerated or simplified features that are especially well suited for comedies such as ***Pocoyo***, ***Open Season***, ***For the Birds***, ***Kung Fu Panda***, or ***Gopher Broke*** (Figs. 4.5.3, 10.4.3, 10.4.4, 10.4.13, and 12.2.12). **Stylized characters** have a consistent treatment of shape—often with both cartoon and realistic elements—and are seen in dramatic or lyrical works such as ***Birthday Boy***, ***A Gentlemen's Duel***, ***Monster House***, ***Ryan***, or ***Même les pigeons vont au paradis*** (Figs. 4.5.5, 10.3.4, 10.3.5, 10.4.5, 10.4.12 and 12.2.3). **Realistic characters** are well suited for creating virtual actors such as Bingo in ***Bingo***, Gollum in ***Lord of the Rings***, ***Beowulf***, Fiona in ***Shrek***, or Gunny in ***Medal of Honor*** (Figs. 4.2.7, 10.3.9, 10.4.16, 11.1.8, 12.1.10, and 12.2.7).

The process of **character development** starts with dozens or hundreds of studies and sketches on paper (Figs. 2.2.1–2.2.5). These drawings can be further developed into small sculptures made with simple modeling materials such as clay or plastiline, an oil-based modeling paste. Once the design is narrowed down and approved, the character is detailed in **character sheets**. These drawings are used to define the main attitudes and emotions of the characters in the form of body positions and facial expressions. Character sheets usually present two different aspects of a character: its anatomy and its personality. In traditional animation, the anatomy of a character is related mostly to the way the character looks. But in three-dimensional computer animation the **anatomy** of a character is related to both its external shape and internal structure. Equally important are the **character turnarounds**, which show the key features of a character from different points of view (Fig. 10.4.2). Character sheets and character turnarounds, both drawn and modeled, are used as templates and guidelines during the character development process and all throughout production.

10.4.3 The personality of these *Open Season* characters and their mixed feelings come across in this particular moment. (Open Season. © 2006 Sony Pictures Animation, Inc. All Rights Reserved. Courtesy of Columbia Pictures.)

External Shape and Silhouette of a Character

The **external shape of a three-dimensional character** defines how the character looks—its visual appearance. Often the shape of a character or the way it moves implies much of the character's personality. For example, think of the difference in motion implied by a fat and heavy body shape with a small round head as opposed to a long and thin body shape with a large cubic head. Or think of the difference in personality that would be projected by a character with a slender frame shaped with a rich assortment of soft curved spaces gracefully connected to each other, as opposed to a disjointed character with a hunched frame covered with unevenly distributed sharp and irregular thorn-like shapes. Of course, appearances can be deceiving, and the personality of a character is not defined just by its shape or its motion. But also keep in mind that much of **casting**—or the assignment of dramatic roles to actors—is partly dependent on the desired appearance of the character to be represented (Figs. 10.4.1–10.4.17).

Animated characters, just like humans, animals, plants, and minerals, come in all shapes. One of the most fun aspects of developing a computer animation is designing the shape of the characters. Character design usually starts as sketches done on paper, and it takes into account the **production technique** that will be used to animate the character. For example, a computer animation developed with limited computing resources may favor characters with simple shapes, while a project developed with unlimited resources may employ

10.4.4 Notice the striking differences in silhouette and personality between the grumpy small characters and the large but affable intruder. The silhouettes are also quite contrasting. (*For the Birds*. © Pixar Animation Studios.)

human-like characters. The design of a character also takes into account the type of story being told and the type of emotions it contains. For example, characters may be caricatures or realistic representations of human beings; each treatment will tell the story in a slightly different way and will also have different animation requirements. Characters may be shaped with stylized forms and ball-joints, or with a single continuous skin. When designing a character one must consider the modeling tools available to do the job, the time constraints, and all the implications of a simple or complex model in the rendering and animation stages. On occasion it is useful to make studies of the character in clay. Three-dimensional clay models complement the character drawings and help to better visualize the overall shading, facial expressions, and body gestures of the characters.

The shape of characters can be used to accentuate an aspect of the character's personality. For example, a big bouncy nose can accentuate the silliness of a character, an overly long tail can be an excuse for its clumsiness, or a slender waist can help focus on its grace and agility. However, it is always the relation between all the parts of a character—and not just a single shape—that expresses the character's personality. When designing a character it is important to consider not only the shape of its body, but also the shape of its clothing. Keep in mind that the realistic animation of cloth may require complex animation techniques.

The **silhouette** of a character helps to define its personality. Shapes with a lot of contrast tend to be more exciting than shapes with no contrast; the latter can even give a static aura to a character.

10.4.5 The lady in *A Gentlemen's Duel* holds her butler tight, while watching her two pretenders fight for her attentions. Physical acting and facial expression bring comedy to this awkward moment. (Created by Blur Studio, Inc.)

Fragile shapes can emphasize a fragile personality or reveal the contrast between a fragile appearance and a strong personality. Heavy and imposing shapes can accentuate the bullying nature of a character or emphasize the contrast between a huge frame and a gentle spirit.

Internal Structure of a Character

Since motion is a fundamental component of animation, the **internal structure of a three-dimensional character** is of the utmost importance in a computer animation because it defines how a character moves. The structure of a three-dimensional character is often determined by its hierarchical **skeleton** and by the functionality of its **joints**. This is convenient since most three-dimensional computer animation systems provide a variety of techniques for manipulating skeletons and joints as if there were muscles moving them. The skeleton and joints of a three-dimensional character are like the frame and joints of a puppet—they define the ways in which it can move. The puppeteer animates the puppet by pulling strings. The computer animator brings the virtual characters to life by manipulating data, and by applying functions and transformations to the skeleton and joints. Hierarchical skeletons are defined in more detail in Chapters 5 and 11, including Figures 5.7.3–5.7.10 and 11.5.1–11.5.5.

The complexity of a character's skeleton and joints controls the timing of its motions (Fig. 9.3.2). A skeleton with a simple structure usually results in simple motions, while complex skeletons may yield complex motions. For example, a character that can move its shoul-

Personality of a Character

External shape
Silhouette
Internal structure
Facial expressions
Body poses
Timing
Rhythm

10.4.6 The main components of the personality of a character.

Drunken Pig
External Shape
Massive, uniform, even boring
Silhouette
Little detail, limbs don't show
Internal Structure
Limited joint mobility
Facial Expressions
None, just a blank stare
Body Poses
Monotonous
Timing and Rhythm
Slow, off-beat, erratic

Perky Girl
External Shape
Well defined, many shapes
Silhouette
Attractive, long limbs
Internal Structure
Lots of mobility, ample motions
Facial Expressions
Varied, multiple, distinct
Body Poses
Wide range, unique, sculptural
Timing and Rhythm
Alive, rhythmic, graceful

Paranoid Guy
External Shape
Simple but varied
Silhouette
Friendly, cartoony
Internal Structure
Very mobile, almost rubbery
Facial Expressions
Limited to eyes, effective
Body Poses
Expressive but small repertoire
Timing and Rhythm
Quick, ahead of the beat

ders independently from the lower torso will be capable of more complex—and expressive—motion than a character that can move its shoulders only in conjunction with the lower torso. Likewise, motions animated with joints that have very limited angles of rotation may be less convincing than motion created with joints that have minimal constraints.

Personality of a Character

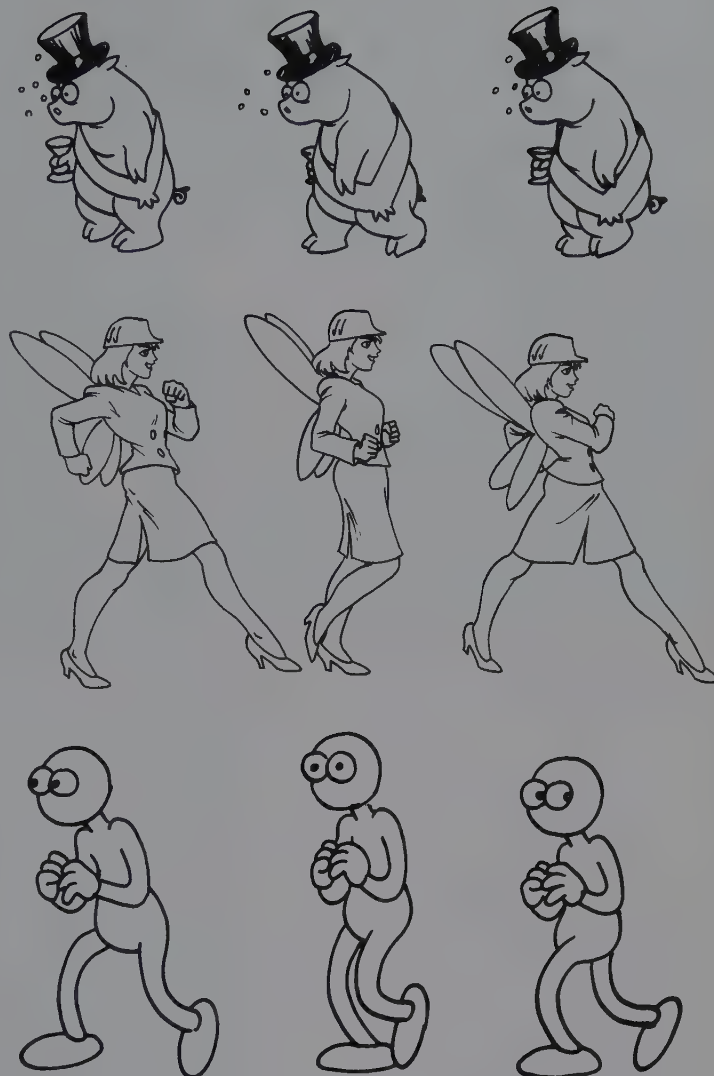
It is usually easier for audiences to follow a storyline when the personality of the characters is well developed and consistent. Audiences get to know a character not only through its dialogue lines but also through its body postures, facial expressions, hand gestures, and walking rhythm (Fig. 10.4.6). It is usually easier to identify with a character that we know, because we can figure out what the character is thinking and predict the action. The personality of an animated character is also defined by many subtle visual elements such as its internal structure, its shape and silhouette, its facial expressions, its timing and its way of moving (Fig. 10.4.7). As mentioned in the previous sections, the skeleton of a computer-animated character helps to define how the character moves and therefore influences its personality. Likewise the silhouette of a character evokes qualities of shapes found in the real world. Figures 10.4.1–10.4.17 illustrate different types of characters with a variety of personalities.

Facial expressions often reveal much of a character's personality and passing emotions. Character sheets include the key facial expressions that define the personality of a character, but it is often necessary to develop hundreds of key facial expressions that will give the character its personality—and that will also be used as keyframes. These facial expressions can be stored in a library of expressions and used throughout the animation. Part of the research for defining those facial expressions consists of observing and drawing the expressions of others as well as making faces in front of a mirror and drawing or modeling them.

The **timing of a character** is essential in defining its personality as well as its emotional state. Timing is about how long a character takes to act or to react, and it includes tempo and rhythm. The **tempo**, or pace, of a motion can have different rates of speed, and these variations in the speed of the character's motion can be very expressive. For example, a slow tempo may express seriousness, fatigue, caution, or intimacy. A flowing tempo may project trust, elegance, or moderation. A lively tempo may express happiness or nervousness. The rate of speed of a character is based on its skeleton but also related to its shape. **Speed** of motion can be constant, can change slowly, or very rapidly. A friendly and trustworthy character may move at a constant speed, while a treacherous and mean character may have sudden changes in the speed at which it moves. The speed of animated objects and characters can also be used to indicate their weight, mass, and power. A fast-moving object implies lightness only if it can

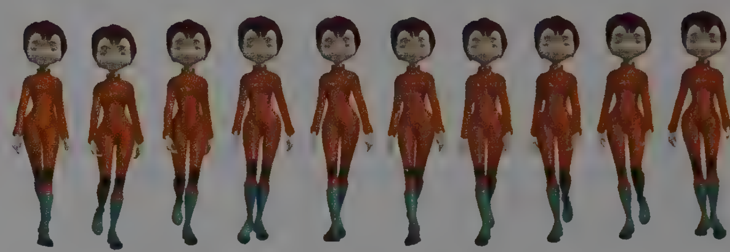
stop relatively quickly. A fast-moving object that requires a long time to stop implies lots of inertia and, therefore, great mass, and weight. The speed of animated objects can also be a good indicator of the amount of stretching and squashing to be expected if and when the object collides and bounces off some other thing in the scene.

The **rhythm** of a character's motion is the repetition or recurrence of motions performed by a character, or the pattern of motion projected by the character. The rhythm of a character's motion can be flowing or broken, regular or irregular. Motion rhythm can combine long motions with short motions, and conveyed by strong and weak beats, in different ways. Think of the differences in personality derived from the rhythm characteristics of three characters as they walk down the road. The first character in Figure 10.4.7, the drunken pig, barely raises its feet off the ground as it walks but keeps its arms close to its stiff torso, and its neck leans forward with the eyesight fixed on the ground. The resulting overall motion—and personality—is quite dull and uneventful. The second character, the perky girl, walks with extreme energy. The head is free and high with a radiant smile on the face. With each step, each shoulder moves back and forth echoing the motion of the opposing foot, the relaxed waist rotates gently from side to side, and the hands hang loose from the arms bouncing back and forth like the shoulders. The motion of the second character speaks for itself, and its personality seems animated and confident. The third character, the paranoid guy, limps as he walks with short, tense steps. Every three or four steps the character stops, nervously turns his head from side to side, and looks around with quick eye movements. His arms are held close to his chest as if seeking some protection.

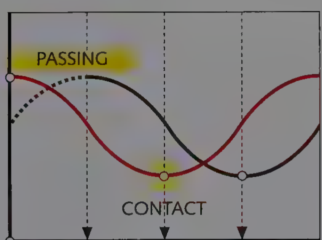


10.4.7 The personality differences of three characters walking down the road are expressed in their motions and body posture, and also listed on the opposite page.

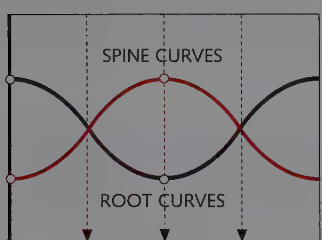
Walk cycles combine physicality, weight and personality, and animating them requires an understanding of body mechanics that—when mastered—can help solve many challenges of physical-based animation. Two general principles that can be tweaked for creating different personalities and degrees of motivation in a walk cycle include splitting the upper torso moves from those of the lower torso, and overlapping the waves of motion that ripple through the body.



10.4.8 Top and side views of a walk cycle. Notice how the hip rotates on all axis, and how it relates to the feet. The hip moves higher on the side of a passing leg, it moves forward as the foot makes contact with the ground and down and sideways right after. (Courtesy of Kyle Balda.)



ROOT Y and Z ROTATIONS



ROOT AND SPINE Y ROTATIONS

10.4.9 Graphs are efficient tools for fine-tuning the broad and subtle motions and overlaps that happen in a walk cycle. (Courtesy of Kyle Balda.)

The source of the locomotion in a walk cycle is in the lower body, in the movements that originate in the hip. This explains why the hip is commonly designated as a character's body center or root. These hip movements include twisting Y rotations that are synchronized with the contact frame of the feet, and tilting Z rotations of the hips during the passing frames (Figs. 10.4.8 and 10.4.9). These combined hip rotations create the internal locomotion that moves the body forward. Each foot absorbs the weight of the body as it touches the ground with each step, and keeps the body standing. As this happens the overlapping motion of the Y translations and X rotations of the hips ripples up the neck and spine.

This wave of lower body overlapping motions determines some of the movement that takes place in the upper body. The torso works to counterbalance the locomotion of the hips with rotations of the spine in the oppossite directions. As the right foot makes contact with the ground, for example, the right side of the hip rotates forward as well. However, the compounded rotation moving up the spine counters the movement of the hips and provides balance and stability to the neck and the head.

Acting, Expressing Emotion, and Thinking Process

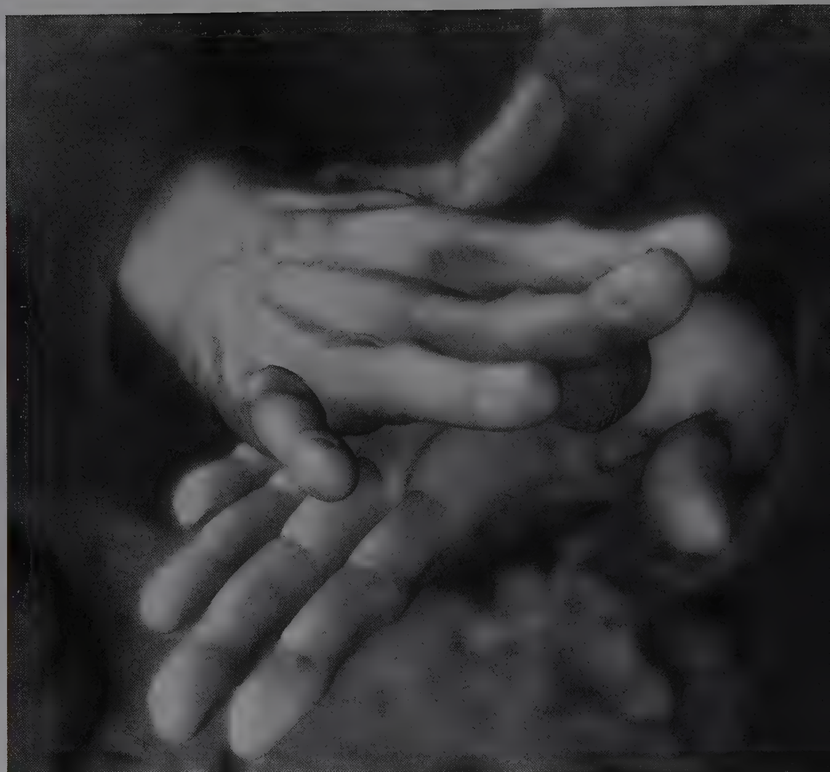
The storyline drives all the events in a computer animation, but the **acting** conveys the emotions and thoughts of the characters. The **emotions** expressed by the character help define the mood of the story. Emotions are a powerful medium for conveying the fine points of a story or for reinforcing some of the character's traits. In addition to feeling emotions, and expressing or hiding them, characters actively think throughout the story. The **thinking process** of characters provides many clues and insights about their fate and the pos-

sible developments of the story. One of the reasons audiences follow the development of a story is because they empathize—or identify—with a character or because they anticipate the action. The emotions, thoughts, and intentions of characters are a perfect vehicle for generating empathy and anticipation in an audience (Figs. 10.4.1–10.4.17 and 12.2.12).

10.4.10 This *Ice Age* moment illustrates how a motion hold allows audiences to focus on what has just happened and think of what is about to happen. Notice how the sloth's IK system allows for extreme bending backwards. (Ice Age © 2002 Twentieth Century Fox. All rights reserved.)

Motions and actions in keyframe animation are often built as a series of poses and gestures that will be used as keyframes to create the motion. When establishing these **key poses** it is important to keep in mind that the goal of the pose is to express an emotion and action and not to just be a beautiful pose. Key poses should not be built with the same criteria that we create, for example, a sculpture. Sculptures are made to be looked at from different angles and throughout long periods of time. Sculptures turn time and emotion into a still pose that we can contemplate. But key poses in an animation are transitory because they are just moments in continuous motion. When composing key poses for keyframe animation it is important to pay attention not only to the visual arrangement of the pose itself but also to the idea or emotion that is being expressed through motions. In computer animations that are based on cartoon characters, **exaggerated gestures** are often used to punctuate dramatic deliveries.

When composing keyframes it is useful to consider the visual line of action. The **visual line of action** determines the position and sequence of the motions in the scene that will guide the eye of the audience to different parts of the image.



10.4.11 The external shape of these two charming characters is soft and playful. Their silhouette and internal structure is simple and straightforward. Their lack of facial expressions focuses the acting of their vivacious and mischievous personalities through their body language. (Agency, Nickelodeon; Director/Designer, Chris Wedge. Courtesy of Blue Sky Productions, Inc.)

10.4.12 (Top right) Expressive hands do tell a story, and smart cinematography includes it as a storytelling accent that complements the facial and body acting. (*Eternal Gaze* © Copyright Sam Chen and Aloha Animation, 2008.)

10.5 Storyboarding

Screenplays are converted into storyboards as they are readied for visual production. A **storyboard** is a visual interpretation of the screenplay and contains many images and production notes. A storyboard consists of a series of panels that contain in visual form the scenes and shots specified in the screenplay. There is no standard medium for storyboards, but they are usually drawn on boards, on plain paper, or preprinted paper with guides. When using preprinted storyboard paper you have to be certain that the proportion of the drawing area corresponds to the proportions of the format that the animation will be recorded on—for example, video, 35 mm film, or 70 mm film. Many characteristics of a storyboard, including its **dimensions**, are determined by whether the main function of the storyboard is to develop the concept, to present the concept to a client, or to guide the actual production of a piece.

The Conceptual Storyboard

A **conceptual storyboard** is used to develop the basic visual ideas such as the actions of the characters, the camera positions, the timing of motions, and the transitions between scenes. Conceptual storyboards

are often loose, sketchy, and informal, and may contain lots of abbreviated notes (Fig. 10.5.1). These storyboards are often drawn on napkins and sketchbooks, and sometimes on letter size or A4 size plain paper.

10.4.13 Even though it is unlikely that a small creature like Shifu could bring someone as large as Po into submission his pose is realistic, with his mass thrown forward and his left arm balancing his weight. (*Kung Fu Panda*™ and © 2008 DreamWorks Animation LLC, used with permission.)

The Presentation Storyboard

The **presentation storyboard** is used to show a detailed visual summary of the project to individuals with decision-making authority, such as clients or supervisors. Presentation storyboards usually include important scenes of the project are often executed with great attention to detail, in color and on high-quality materials (Fig. 10.5.3). The visuals are usually large enough so that several people in a meeting room can look at them from a distance, and small enough to fit inside a portfolio. The notes included in presentation storyboards should always be very legible and descriptive without getting too technical.

The Production Storyboard

A **production storyboard** often guides the production of an animated project or visual effects shot (Fig. 10.5.2). This type of storyboard can be the document that everybody involved in the production process refers to, to clarify questions. For this reason, they are always very detailed and precise, and they include drawings and written information about every shot in the story. It is very important to work out many of the technical details in an animation before a production storyboard is created. Otherwise, information contained in the storyboard may change a lot, rendering it useless for production purposes.

The **written information** in production storyboards may include detailed descriptions of the motion, the camera, the set, the lighting, other rendering specs, the timing, and the transitions

10.4.14 The sassy personality of this bird is evident in his pose and facial expression. (*Occho Kochoï* © 2006 TeamTO–TF1–Teletoon.)

10.4.15 The impatient facial expression and the menacing body language of Lord Farquaad make his intentions crystal clear. (*Shrek*™ and © 2001 DreamWorks L.L.C.)

10.4.16 Two core poses of Gunny, the *Medal of Honor* gunnery sergeant: pistol standing attack and a 20–30 frame run cycle (opposite page bottom). The animation rig for this character is shown in Figure 5.7.3. (© 1999 Electronic Arts Inc. All rights reserved.)

between the shots. Production storyboards also include soundtrack information such as transcripts of the dialogue and narrator's voice-over, and descriptions of the music and sound effects.

Each still frame that represents a shot in the storyboard is usually numbered with a **shot number** or a **scene-and-shot-number**. The timing information for each shot is usually noted right below the visual representing each shot. Absolute timing values, or **elapsed time**, indicate the exact time at which the shot starts and ends in terms of hours, minutes, seconds, and frames. For example, a one-and-a-half-second-long shot in the middle of the storyboard for a short video animation may start at 00:01:37:15 and end at 00:01:39:00. Relative timing values or **running time** simply indicate the total length of a shot, usually in seconds and frames; for example, the same shot may be one second and 15 frames (1 sec. 15 fr.) long.

The **drawings** in the storyboard depict the images seen by the camera. Sometimes the motions of the camera are further clarified by using directional arrows to indicate them visually over the drawing. The points of visual interest in the composition and the direction of the paths of visual interest are sometimes also overlaid on the still drawings in the storyboard (Fig. 2.7.5).

The passage from one shot or scene to the next is called a **transition**. Some of the most common transitions between shots specified in a storyboard include a cross-dissolve, a fade-in, a fade-out, or a soft-cut (these are explained in Chapter 14). Plain cuts between shots are usually not indicated in storyboards because it is assumed that most of the transitions are cuts. Aspects of the camera specified in a storyboard include the type of shot, the type of move, the point-of-view or POV, and the type of lens (see Chapter 11 for more information on camera animation). Production storyboards can

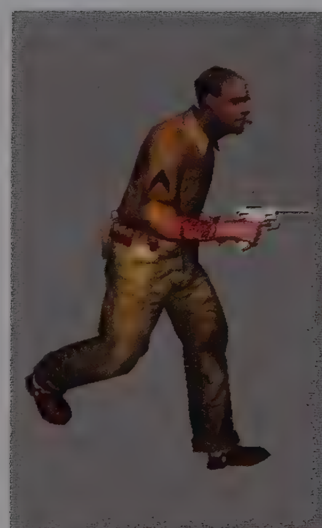
be drawn in formats that can be easily pinned to a wall or carried in a binder or briefcase. Sometimes these storyboards are drawn so that the visual and written information on each shot is contained on a single piece of paper. This way a sequence of shots in a storyboard can be easily rearranged if necessary while production is underway. Other times these storyboards are drawn on letter size or A4 size paper so they can be carried by different members of the animation team.

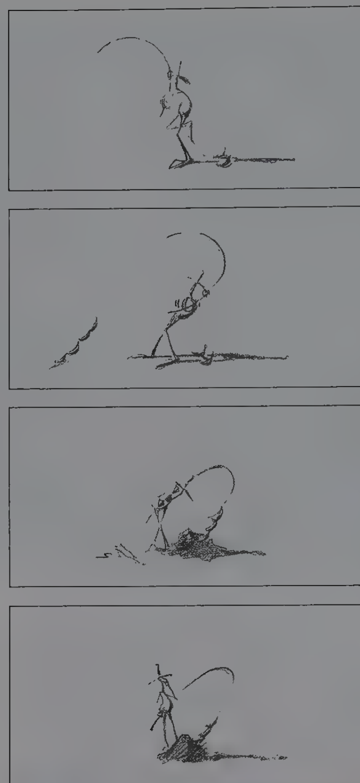
10.4.17 The protagonist in *Planet 51* features cartoon proportions with a realistic finish. (© Ilion Animation Studios/HandMade Films International. All rights reserved.)

Titles and Credits

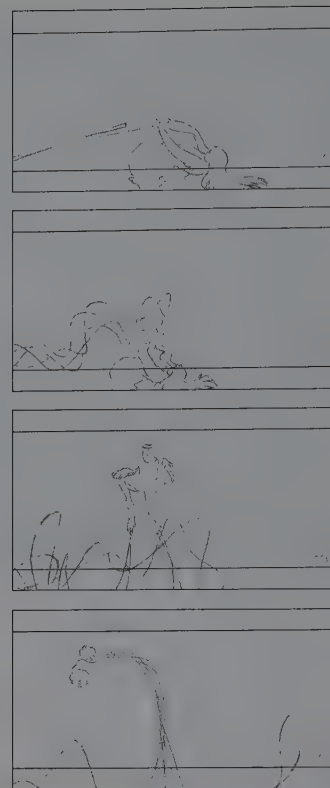
In addition to the animation itself, many production storyboards commonly contain detailed information about the titles, the opening and closing credits, and any other text, letterforms, or graphic information that may appear in the animation. The titles of a computer animation can be simple or elaborate, but they are always storyboarded in detail. Simple titles typically consist of two-dimensional letterforms superimposed on the animated opening sequence (Fig. 10.5.4). Two-dimensional letterforms are usually created with a **character generator** or an electronic **paintbox**, and not with a three-dimensional modeling program. The latter is used in cases when the titles are of three-dimensional nature or when they are part of the animated environment. The credits almost always consist of two-dimensional letterforms. Some major credits may be superimposed on the opening sequence, but most appear at the end of the animation as **rolling credits**. (See "Creative, Technical and Production Teams" in Chapter 2 for more information about the customary way to credit those who participate in a computer-animated project.)

When designing the placement of text and graphics on the screen it is important to make sure that they are readable and that

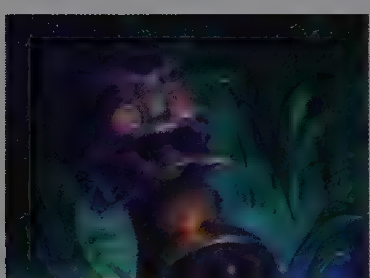




10.5.1 (Near right) A conceptual storyboard indicates the essence of a shot or a sequence without getting into the details. See the rendered version in Fig. 1.4.9. (Images from *Fishing*, a PDI short film by David Gainey.)



10.5.2 (Far right) Frames from the *Shinobi* production storyboard showing the detailed positions choreographed for the human actor, its virtual stunt double, and the digital clothing and extensions. (Images courtesy of Links DigiWorks Inc. © 2005 SHINOBI Film Partners.)



10.5.3 (Above) Detailed renderings are commonly used in presentation storyboards. (© 2003 Oddworld Inhabitants, Inc. All rights reserved.)

10.5.4 (Top right) The opening title for *A Gentlemen's Duel* uses a type style from the period in which the story is set. The main location is in the background. (Created by Blur Studio, Inc.)

they will not be cut off by being too close to the edge of the frame. **Field guides** are graphs with concentric rectangles that can be used to specify the exact position of text and graphics within the frame (Fig. 7.2.2). The **title safety area** in the field guides clarifies what constitutes a placement of the titles and graphics that may not always be within the frame due to slight differences in vertical and horizontal positioning of the image among television sets or film projectors.

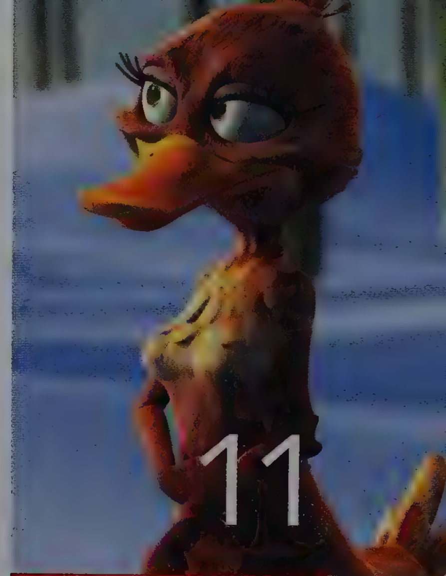
CHAPTER 10

Key Terms

24 frames
30 frames
Acclaim AMC, AMF formats
Act
Acting
Ahead of the story
Anatomy
Animatronics
Animé, anime
Anticipation
Arcs
Attention of the audience
Background
Beginning of the story
Behind the story
Biovision BVA, BVH formats
Branching
Cartoon characters
Cartoon character animation
Casting
Cel animation
Cels
Character animation
Character appeal
Character development
Character generator
Character sheets
Character turnarounds
Clean-up drawings
Clear delineation
Complex motion
Conceptual storyboard
Dialogue
Digital backup
Dimensions
Drawings
Effects animation
Elapsed time
Emotions
End of the story
Events
Exaggerated gestures
Exaggeration
External shape of a three-dimensional character
Extremes
Facial Action Coding System, FACS
Facial expressions
Fake hold
Fast-in, fast-out
FBX file format
Field guides
Flow of events
Flowcharts
Follow-through action
Foreground
Forward kinematics
Frame
Frames per second, fps
Hand-drawn animation
Hierarchy level
Hybrid framework
Implementing the motions
In-between
In-betweening
Ink and paint
Intended audience
Interactive project
Interactivity
Internal structure of a three-dimensional character
Interpolation
Interrupted action
Inverse kinematics
Joints
Key poses
Keyframing
Keyframes
Layers of motion
Linear
Manual arrangement
Middle of the story
Montage
Motion capture
Motion control techniques
Motion dynamics
Motion hold
Nonlinearity
Overlapping action
Overlapping motion
Paintbox
Parallel action
Pencil test
Performance animation
Plot points
Pose-to-pose action
Presentation storyboard
Primary motion
Procedural motion
Production storyboard
Production technique
Puppetry
Rate of display
Readability of motion
Realistic characters
Rhythm
Rolling credits
Rule-based motion
Running time
Scene
Scene-and-shot-number
Screenplay
Scripts
Secondary action
Secondary motion
Sequence
Sequences of still images
Shooting on twos
Shot
Shot number
Silhouette
Simple motion
Skeleton
Slow-in, slow-out
Solid drawing
Solid modeling
Speed
Squash and stretch
Staging
Stand-alone
Still frame
Stop-motion animation
Stories
Story beats
Storyboard
Straight-ahead action
Structure
Stylized characters
Subject of a screenplay
Tempo
Thinking process
Timing
Timing of a character
Timing of the action
Title safety area
Transition
Translated into moving images
Treatment of the subject
Twelve principles of animation
User testing
Visual effects animation
Visual interpretation
Visual line of action
Walk cycles
Written information

CHAPTER 11

Computer Animation Techniques



QUICK INDEX

Summary

Keyframing and forward kinematics are useful basic techniques for animating the position, orientation, shape, and attributes of three-dimensional characters. In addition, using parameter curves to animate provides a powerful tool to sketch out and fine-tune character animation. A variety of other computer animation techniques are also reviewed in this chapter, including shape animation of three-dimensional models with lattice deformation or morphing; and the interpolation of attributes like the surface characteristics of models, the depth of field of cameras, and the color of lights. The chapter concludes with a review of basic hierarchical rigs as they relate to character animation, and the integration of two- and three-dimensional computer animation.

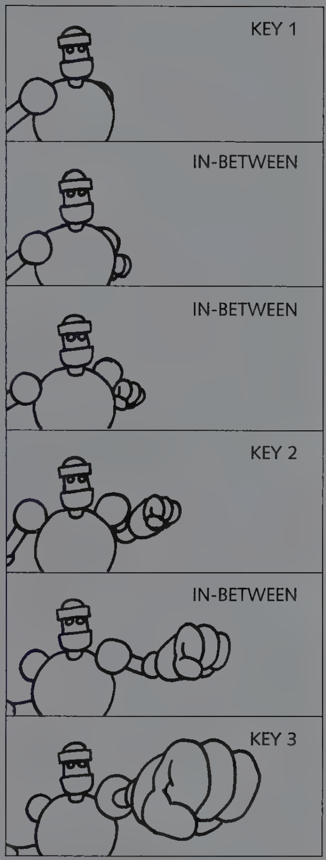
11.1 Keyframe Interpolation and Parameter Curves

As described early in Chapter 10, defining keyframes with key poses and creating the in-between positions is still a hallmark of animation. Using a technique called interpolation computer animation software can create the in-between positions. This is called **keyframe interpolation**, and it is an essential computer animation technique used in a myriad of ways to create sequences of still frames. Keyframe interpolation calculates the **in-between frames** by averaging the information specified in the keyframes (Fig. 11.1.1). Interpolation techniques can be used to calculate the position of objects in space, as well as their shape and other attributes. The most common types of interpolation include linear and curve interpolation.

Establishing a **keyframe** is usually done interactively, using an animation **timeline** that stores the position and duration of the different moments and actions being animated. Interpolating keyframes allows us to control the time, or **speed**, that it takes to get from one keyframe to another and the **rate of change** between frames (Fig. 11.1.2).

Interpolation is commonly expressed in the form of a graph

11.1.1 (Opposite page) This 18-frame forward kinematics animation cycle shows the real-time model of *Spyro the Dragon*™ preparing to blow a flame of fire, blowing it, and returning to the resting position. (Spyro the Dragon™. Courtesy of Universal Interactive Studios, Inc. and Insomniac Games.)



11.1.2 A sequence with three keyframes and two interpolations—one quicker than the other.

(Previous page: *The Ugly Duckling and Me*. © 2006 A. Film, Magma, Futurikon, Ulysses.)

that shows the relation between time and the parameter being animated. Time is usually represented by the horizontal axis, and the parameter in question is usually represented by the vertical axis. The **slope** of the path in the graph represents the speed or rate of change. A flat path, for example, indicates no change—zero speed—a diagonal path indicates constant change, and a curved path represents variable change. The steeper the slope of the path the greater the rate of change (Fig. 11.1.3). Interpolation graphs are generated automatically by most computer animation software as soon as the animator specifies the animation parameters on one or several objects in the scene. It is usually possible to edit these graphs interactively.

Linear Interpolation

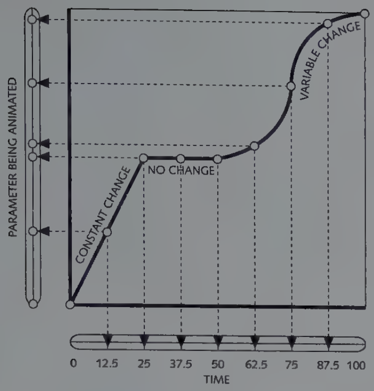
Linear interpolation is the simplest and most straightforward computer animation technique for calculating in-between frames. **Linear interpolation** simply averages the parameters in the keyframes and provides as many equally spaced in-between frames as needed. However, linear interpolation techniques may produce mechanical results when applied to subtle motions unless a significant amount of work and animation skill are used to fine-tune the results. Linear interpolation is based on constant speeds between the keyframes, but it produces abrupt changes in speed on the keyframes where one constant speed ends and a different one starts. **Constant speed** is represented by the straight lines in the graph. Linear interpolations cannot handle subtle changes in speed because the in-between frames are created at equal intervals along the path (Fig. 11.1.4).

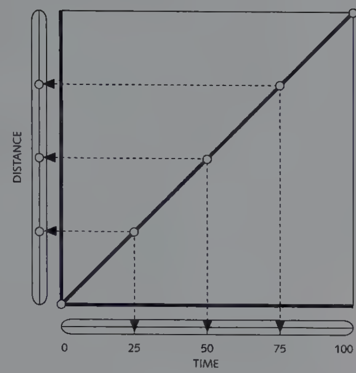
Curved Interpolation

Curved interpolation, also called an **interpolation ease**, is a technique for calculating in-between frames that is more sophisticated than linear interpolation. Curved interpolation averages the parameters in the keyframes, taking into account the variations of speed over time, known as **acceleration**. When curved interpolation is represented in graphical form the increase in speed, also called an **ease in**, is represented by a line that curves up. An **ease out**, or decrease in speed, is represented by a line that curves down (Fig. 11.1.5). Therefore, the distribution of in-between frames along the path depends on whether the rate of change increases or decreases. Curved interpolation can also include motion with constant speed, and that is represented with straight lines (Fig. 11.1.4).

Working with Parameter Curves

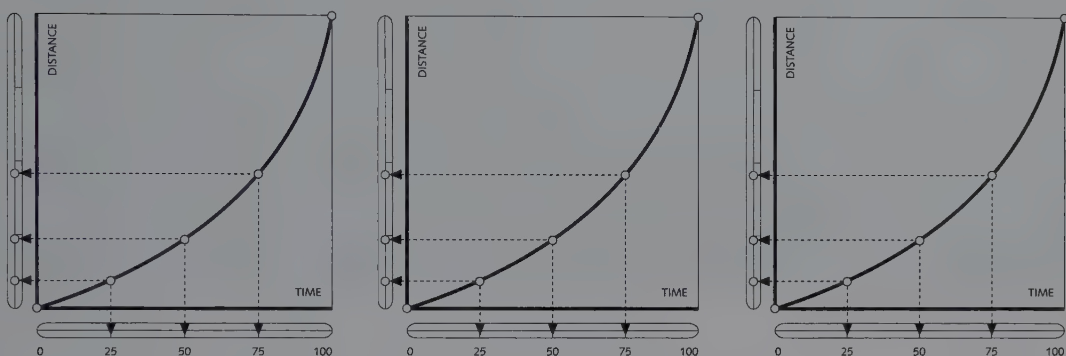
A graph representing curved interpolations is also called a **parameter curve** or a **function curve**. These graphs are generated automatically by computer animation software as the animator moves, for example, an entire character or parts of the character. Working with





11.1.3 (Far left) A changing slope in the Time/Distance interpolation path results in changes of speed, acceleration, slow-ins and slow-outs.

11.1.4 (Near left) A linear interpolation graph using time and distance as the parameters. The diagonal line represents constant speed.

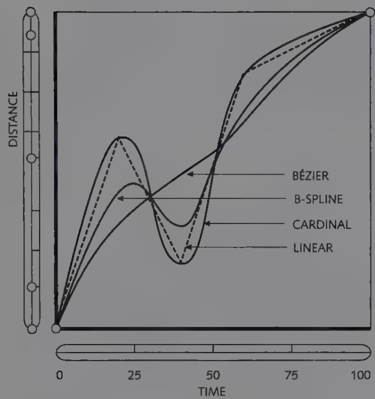


11.1.5 Three graphs for curved interpolation. Notice how the distance traveled over time by the animated model varies with the slope of the curve.

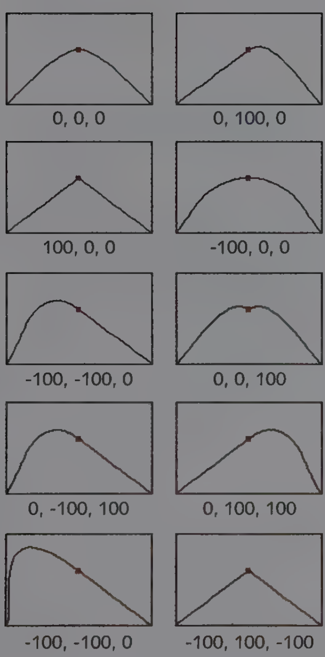
parameter curves provides animators with an additional method for modifying the animation. Going back and forth between interactive manipulation of the animation rig and working with parameter curves is a common way to edit and fine-tune subtle aspects of an animation. Parameter curves can be generated for each joint in a rigged character.

The exact shape of a function curve depends on the type of curve used (Fig. 11.1.6). Some programs provide interpolation graphs with all types of curves. As mentioned in Chapter 4, some of the most popular types of curves include linear splines, cardinal splines, b-splines, and Bézier curves. Each of these types of curves is shaped in a characteristic way by their control points, or control vertices.

The animation generated by linear splines, for example, is based on a constant speed between keyframes and abrupt changes of speed at the keyframes where two different constant speeds meet. The animation generated with a cardinal spline is very dependent on the placement of the keyframes in the interpolation graph because the control points in a cardinal spline force the curve to pass through all of them. The tight fit of cardinal curves to the control points often results in harsh closed curves, which translate into rough eases. The animation created with b-splines usually contains smooth interpola-



11.1.6 Four types of function curves with the same control hull have been overlapped to show the different graph results.



11.1.7 Motion paths can be eased in and out with sliders that control the tension, bias, and continuity. These function curves illustrate a variety of settings.

tion eases because the shape of the curve is loosely controlled by the control points but not forced to pass through them. Parameter curves calculated with Bézier curve functions offer a flexible and complex control path because the shape of the curve is controlled by the position of both the control points and the tangent points. Bézier curves are flexible curves that, for example, can blend softly or sharply into a straight line. Another advantage of using Bézier curves for defining interpolation graphs is that their slope can be modified with just the tangent points, therefore avoiding the insertion of additional control points (keyframes) to shape the curve and the motion.

Parameter curves can represent either a simple linear interpolation, an ease in, or an ease out. But the curves that include the three interpolation types are called **ease functions**. These complex interpolations can be defined interactively, for example, by dragging a slider that represents the proportion of ease in, linear, and ease out in the function (Fig. 11.1.7). Ease functions can also be specified by inputting numerical values that range from 0 to 100 percent or from 0 to 1 (Fig. 11.1.9).

Interpolation of Position and Orientation

Interpolation techniques can be used to calculate the position and orientation of animated objects in three-dimensional space. This includes the models in a scene as well as cameras and light sources. As mentioned in Chapter 3, the position and orientation of an object in three-dimensional space can be modified by changing the values in the **transformation matrix** that controls the translations, rotations, and scaling that are applied to the objects in the scene. When keyframe interpolation techniques are used, the values in the transformation matrices of moving objects are specified at the keyframes, and the in-between values are interpolated. Editing parameter curves provides animators with an easy way to edit the transformation matrices of a complex object.

Interpolation of Shape

Interpolation techniques can also be used to animate the shape of three-dimensional models. The basic idea behind **shape animation** consists of transforming one key shape into another one by letting the interpolation techniques calculate all the in-between positions of the points and lines that define the shape of the model. A variety of shape interpolation techniques were used to create the images in Figures 11.1.8 and 11.2.6.

Interpolation of Attributes

The **attributes** or characteristics of models, cameras, and lights—other than spatial position and shape—can also be animated with interpolation techniques. In the case of three-dimensional models, it

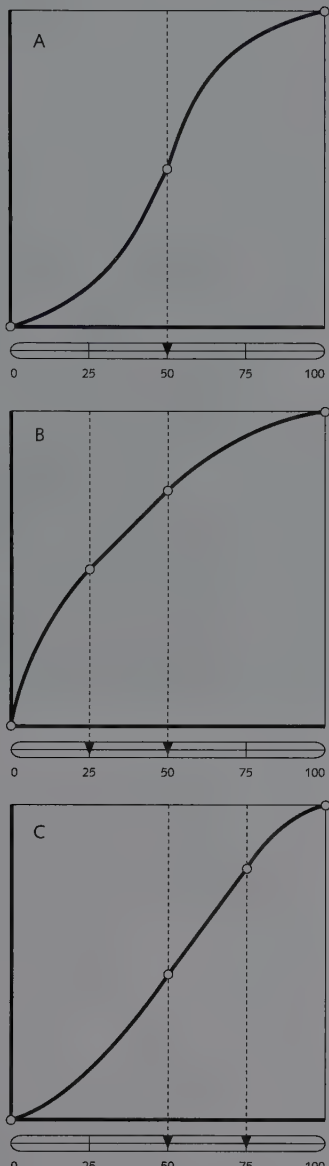
is common to animate their surface characteristics, such as color, texture, or transparency. Focal length and depth of field are two characteristics of cameras often animated. In the case of light sources, it is common to animate attributes such as color, intensity, cone-angle, and fall-off values. Several interpolations of attributes are covered in the next three sections in this chapter, and texture interpolation is illustrated in Figure 11.2.9.

11.1.8 The pose of Princess Fiona from *Shrek* is defined by interpolation of shape, position and orientation, geometry, and range of motion, but also by the comedic moment. (*Shrek*™ and © 2001 DreamWorks L.L.C.)

11.2 Forward Kinematics and Model Animation

Keyframe interpolation techniques, and manipulation of parameter curves, are effective for animating the position, shape, and attributes of three-dimensional models. Forward kinematics is possibly the most common of all basic techniques for animating the position of rigged characters (not including, of course, inverse kinematics). The shape of models can be animated with a variety of techniques that include free-form shape interpolation, three-dimensional morphing, and external control structures such as lattices and functions. Spectacular results can be achieved by animating the surface characteristics of models with interpolation techniques. All these techniques are based on the principle of keyframe interpolation.

The most common method for specifying the **spatial animation** of three-dimensional characters when using computer-based keyframe interpolation is to interactively arrange the position and



11.1.9 Ease parameter curves specified with different numerical values: 50% ease in and 50% ease out (A); 25% ease out, 25% linear, and 50% ease out (B); 50% ease in, 25% linear, and 25% ease out (C).

orientation of their parts. This is achieved by defining interactively key poses on the timeline while manipulating the geometry and rig in three-dimensional space, previewing the results and refining the motion with the parameter curves, and by letting the software interpolate the in-between frames.

Forward Kinematics

Forward kinematics, also known as **FK**, is a useful technique for specifying motion of jointed characters through key poses. It is a simple technique that requires a great deal of manual work, unlike the technique of inverse kinematics described in Chapter 12, but it offers almost absolute control of character poses and moves. **Interactive keyframing** with forward kinematics consists of determining the motion and final position of a model by specifying the new position of each and every joint that moves in the keyframe (Figs. 11.1.1 and 11.2.1). This can be accomplished by typing the numerical value for each joint angle directly into the appropriate fields or dialog boxes provided by the software (Fig. 4.6.1), or interactively, by using a mouse, graphics tablet and pen, track ball, or input peripheral like the one pictured in Figure 11.2.2. Forward kinematics is best used in situations when the straight-ahead animation style is sought.

The project illustrated in Figure 11.2.3 shows the results of a procedural animation system that uses a combination of forward kinematics expressed in the form of key actions, body constraints, and a small amount of random noise to drive the figure. This particular program seeks to create interactive animated characters that dance with some emotional expressiveness and a minimum number of joints. In this case, the joints were universal—four for each limb and one for each waist, neck, and head. Even with such a small number of joints, this figure has two separate joints at the base and the top of the neck because the head position is so important for emotional expressiveness. In this example, the actions are "taught" to the animated character in advance, and the transitions between actions are adjusted so that the motion looks natural. Establishing the proper joint angles before the motions occur is important since most of the personality of the character and the mood of the dance step is expressed through body language. The model of the dancer is built with ellipsoids. Some of the motion constraints include a few simple joint constraints—the head, for example, cannot turn all the way around. Obstacles in the path are avoided by turning away from them, and the supporting foot at floor level propels the character. Predefined actions such as a dance step can be stored in the form of a table of ranges for each joint involved in the action. The values used to specify the rhumba dancer are also listed in Figure 11.2.3. The lowercase letters represent a variety of functions, including a random noise generator function (n) applied to the head to simulate the character looking around as she dances. The arm joints at the chest (Rchest and Lchest) are controlled by functions in the X and Z axes that result in an ellipti-

cal motion of the shoulders. The noise function is also used to give the standing-still figure of the dancer a subtle restlessness and apparent shifting of weight back and forth from one side to the other.

Motion Paths

The technique of **motion paths** provides an additional method for animating characters and objects in three-dimensional space. The characteristics of the motion path, usually drawn as a single path in the three-dimensional environment, can also be controlled with parameter curves.

Three-dimensional motion paths are easy to work with because they allow the animator to quickly rough out motion that may involve multiple translations and rotations. Working with motion paths allows us to think of motion in terms of "go from here to there, and follow this path." This technique can be used to control many types of motion. It is especially useful, for example, to define with precision the motion of flying models or cameras or objects that move by sliding over a surface (Fig. 11.2.4).

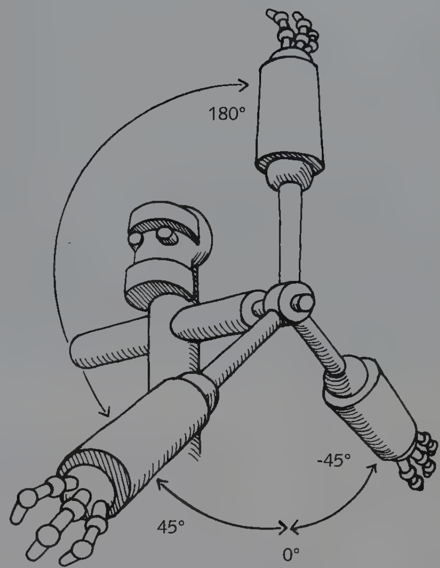
Motion paths are also useful to sketch or lay out the motion of traction vehicles like bicycles and cars, and also of legged creatures like ants and humans. In these cases the motion path is used to define the primary motion with the secondary motion layered over and animated with other techniques such as interactive keyframing, inverse kinematics, or motion dynamics.

Motion path animation is defined in several steps, and it starts with a curve being drawn in three-dimensional space. Then the model to be animated is selected and **linked to the path**. The timing parameters of the path are then defined in the timeline. This is usually done by typing the values of the frames in which the motion path animation is to start and end. When a motion path animation is first defined, a linear interpolation with constant speed is often applied as a default value. But the speed and acceleration can be easily changed by editing its timing with the parameter curves.

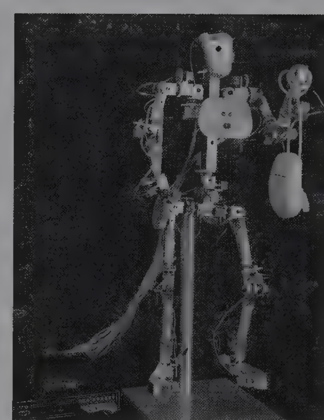
Most software programs will keep the object linked to the motion path so that the front of the animated object is always facing in the direction of the path (Fig. 11.2.5). This is also called keeping the direction of the object tangential to the path. In order to do this, most computer animation programs need to know which is the front of the object at the time when the object is linked to the motion path.

Free-Form Shape Animation

Simple shape animation can be created by interpolating the shape of two objects on a point-by-point basis. When this interpolation is done between two versions of the same geometry it is usually called free-form shape animation, and when it is done between two different models, it is called morphing. **Free-form shape animation** can be created by establishing a few keyframes of the same geometry, polyg-



11.2.1 In forward kinematics motion is created by specifying the angle of a joint. In this case a total range of motion of 225 degrees (45, -45, and 180) is shown.



11.2.2 This articulated model can be used to input and visualize joint angles to animations based on forward kinematics techniques. (The Monkey™ is courtesy of Digital Image Design.)

Forward Kinematics Procedural Animation

```
{
  {5 5 5}      {-5 -5 -5}    {n1 n2 n3}  Nod
  {15 5 0}     {-15 -5 0}    {b a}       Rchest
  {0 0 0}      {0 0 0}       {a a}       Rshoulder
  {-90 0 0}    {-70 0 0}     {a a}       Relbow
  {0 0 0}      {0 0 0}       {a a}       Rpelvis
  {-25 5 -15}  {0 -10 0}     {a a a}     Rhip
  {50 0 0}     {0 0 0}       {a a}       Rknee
  {0 0 0}      {0 0 0}       {a a}       Rankle
  {0 10 0}     {0 -10 0}     {a a}       Waist
  {-15 -5 0}   {15 5 0}      {b a}       Lchest
  {0 0 0}      {0 0 0}       {a a}       Lshoulder
  {-70 0 0}    {-90 0 0}     {a a}       Lelbow
  {0 0 0}      {0 0 0}       {a a}       Lpelvis
  {0 -20 0}    {-10 20 -25}  {a a a}     Lhip
  {0 0 0}      {20 0 0}      {a a}       Lknee
  {0 0 0}      {0 0 0}       {a a}       Lankle
} 'rhumba define_action
```

11.2.3a Program with commands for a virtual dancer to dance a rhumba sequence in *Danse Interactif*. (Courtesy of Ken Perlin, New York University, Media Research Laboratory. © 1994 Ken Perlin.)

onal or spline-based, and progressively modifying the shape of the models. This can be done by pulling the points in its mesh one-by-one or in clusters. The in-between frames interpolated by the software constitute the shape transformation.

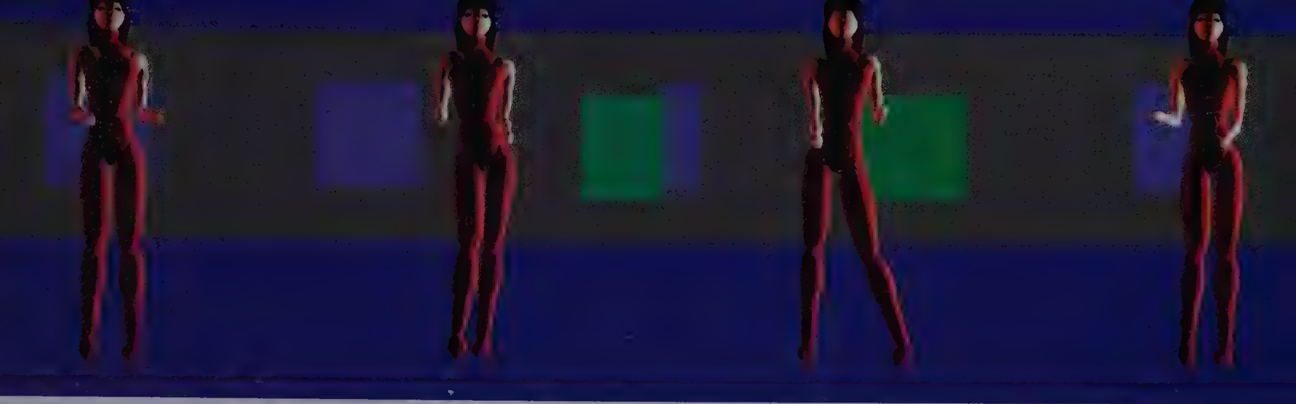
The free-form shape animation process starts by identifying the points—or control vertices—in the model that will be animated. This is usually done in the wireframe mode as more points are visible without having to constantly turn the model around. A single point or a group of points are selected and dragged to a new position. Free-form shape animation can be useful in the creation of squash and stretch effects. **Squashing** and **stretching** are commonly used in animation to emphasize the motion of objects in response to the forces of compression and expansion. Squash and stretch effects help characterize the mass and weight of moving objects as well as the material they are made of.

Three-Dimensional Morphing

Three-dimensional morphing is an effective technique for creating shape animations of two different geometries. **Three-dimensional morphing** works by animating all the points of one object into the positions occupied by the points of another (Fig. 11.2.6). The results of three-dimensional morphing animation are usually fascinating, but there are two important technical requirements that must be satisfied before this technique works to its fullest. First, the best results are obtained when each of the three-dimensional models has the same number of points. This fact implies that a fair amount of preplanning—especially during the modeling stage—is necessary for this technique to be practical. Many software programs will not even attempt three-dimensional morphing unless this condition is satisfied. Second, it is also necessary to specify the **order of correspondence** between the points in each of the three-dimensional models. Many software programs allow for the interactive linking of points between objects. This is helpful to make sure that the morphing results do not include undesired accidents such as objects that wrap inside out, overlapping surfaces, or holes in the models. In some cases, however, these shape accidents may be appropriate effects to help tell a particular story.

Free-Form Lattices

Free-form shape animation can create striking results, but it requires a great deal of skill and time to manipulate points. Using **external control structures** to regulate the shape animation of objects can be a better choice, especially in cases where a uniform shape deformation is desired. Two popular animation techniques that use external control structures are free-form lattices and wave functions. A **free-form lattice** is a three-dimensional grid of control points and lines that manipulates the points in a three-dimensional model as if the control



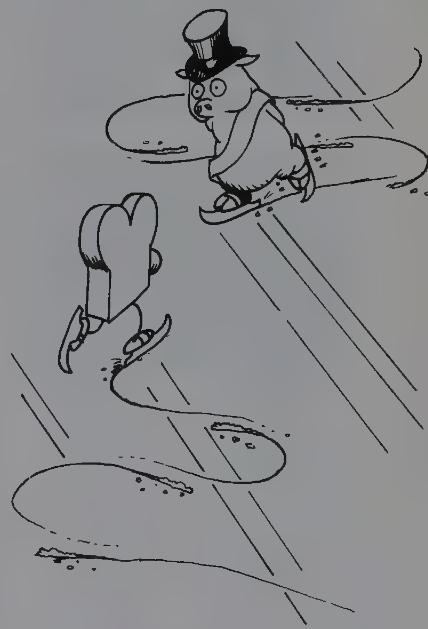
11.2.3b Still frames from *Danse Interactif*, a real-time procedurally animated interactive dance performance. These four images represent the cycle the animated dancer goes through in completing the rhumba move. (Courtesy of Ken Perlin, New York University, Media Research Laboratory. © 1994 Ken Perlin.)

points in the lattice were connected to the points in the geometry with springs. As the control points in the lattice are moved they push or pull the points in the object (Fig. 4.5.8). The ability to create shape animations by moving one or several points in a free-form lattice is directly related to the resolution of the lattice. A lattice with only a small number of points yields rough shape animations, while lattices with larger numbers of control points can be used to apply more subtle local distortions on the model controlled by the lattice.

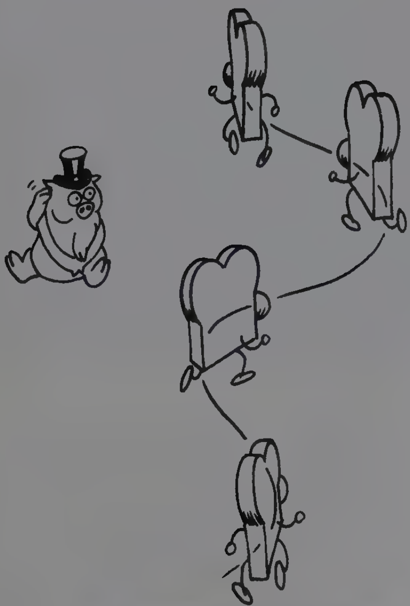
Wave Functions

A good number of mathematical functions can be applied to a three-dimensional model with the purpose of changing its shape. Animating with functions can be an economical way to create unusual animations because little hand work is required once it has been determined how to apply the function to the object (Fig. 11.2.7). For most animations that involve traditional storytelling, animating with functions provides an effective way to create the foundation of a motion that can be complemented with other techniques. Most functions tend to be of limited use and are rarely used as the primary technique in a computer animation project. This is because the motion generated with functions—while interesting and exciting from a mathematical point of view—is usually not suitable to be used by itself, either because it is too simple or it calls too much attention to itself.

Two-dimensional function curves can be used as control structures to animate the shape of objects. These control curves are often called **wave functions** since their shapes resemble the outline of a wave. (Procedural textures created with a multiple wave function are shown in Figure 9.6.4.) The distortions created by applying wave functions to three-dimensional models are sometimes unexpected, and a trial-and-error approach is often required. Some of the characteristics of this technique, however, are quite simple and easy to control (Fig. 11.2.8). The wave type variable determines the way in which the wave propagates from its center throughout the three-dimensional model. Functions can be easily looped and used to simulate the recurring motions, for example, of water waves. **Circular waves** are an excellent technique for recreating the motion of the



11.2.4 The motion paths of two skaters are made visible here.



waves on the surface of a lake, and **planar waves** can recreate the effect of waves on the surface of the sea. **Spherical waves** can be used to recreate an explosion.

Animation of the Surface Characteristics

Changes in color, transparency, texture, or reflectivity often indicate a change in the inner emotions, chemical composition, or state of mind of objects or characters. These changes in the surface characteristics can be useful elements of visual storytelling because they happen as responses to internal or external factors.

Some changes in surface characteristics happen rapidly in reality—in a matter of seconds—while others require several months or years to occur. Both the blushing of a face or the maturing of an apple, for example, involve changes in color and visual texture, but one happens in seconds while the other requires weeks. In both of these cases, the **timing of the transformations** of color and texture is crucial to the understanding of the action. If the reddening of an apple takes place throughout a long period of time we know that we are watching a **natural transformation**, but if it happens in a matter of seconds we assume that some **fantastic transformation** process is under way. Likewise, if we watched the blushing occur in a matter of seconds we will know that somebody is expressing sudden feelings of shame, excitement, or modesty. But if the reddening takes place throughout several days then we would assume that transformation to be the reaction of the skin to an allergy or disease.

Animating the surface characteristics of objects or characters can create realistic or fantastic effects of material transformation and also communicate subtle or sudden changes of events in a story. The animation of surface characteristics is easily done with interpolation techniques. The process is simple but powerful. A set of surface characteristics is applied to the objects in question in the first keyframe of a sequence and a different set of surface characteristics is applied in the last keyframe. The surface parameters can be easily edited in the relevant dialog boxes by typing the appropriate values. Figures 11.2.9 and 11.2.10 illustrate the animation of a variety of surface characteristics.

In addition to animating the standard shading attributes of a three-dimensional surface, it is also possible to create spectacular effects by mapping sequences of animated images on a three-dimensional surface and by animating the parameters of a three-dimensional procedural texture. **Mapping a sequence** of images is done by assigning two-dimensional picture files that are applied as maps to three-dimensional objects (Figs. 9.3.3, 9.5.1, and 9.6.2). The sequencing can be done on a one-to-one basis (one two-dimensional image per frame of three-dimensional computer animation) or by following a script where all image files in a folder, for example, may be mapped in sequences onto animated geometry. See Chapter 9 for more information about surface characteristics and their variables.

11.2.5 These cameras look down the motion path as they move. (*Medal of Honor* screen shots. © 1999 Electronic Arts Inc. All rights reserved.)

11.3 Camera Animation

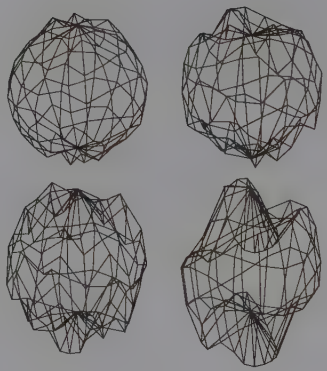
The camera plays an important role in computer animation because its motion and the changes in some of its attributes can have a powerful storytelling effect. As explained in Chapter 7, the point of view of a camera and the type of camera shot are both defined by the position and orientation of the camera. All camera motions also require a change in the position and orientation of the camera.

The motions of the virtual cameras used in computer animation are based on the camera moves defined in traditional cinematography. Most software programs use the same camera names as in traditional cinematography, but some use a slightly different nomenclature. All the possible camera moves can be expressed in terms of translations or rotations around one or several camera axes. In addition to changing the position and orientation of virtual cameras, their focal length and depth of field are some of the attributes that can be easily animated.

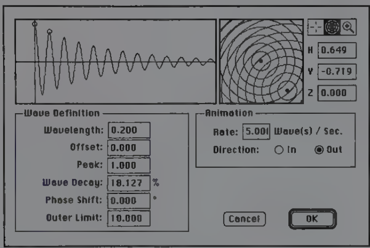
Position Camera Moves

The position of a camera can be easily defined by typing an **absolute position** value specified in XYZ world coordinates in the field or dialog box that controls the camera position. This technique can be useful for defining **locked shots**, where the camera must be

11.2.6 This still frame from the film *The Mask* is an example of three-dimensional morphing. (© 1994 New Line Productions, Inc. All rights reserved. Courtesy of New Line Productions, Inc.)



11.2.7 A circular sine function was used to distort a sphere.



11.2.8 This dialog box shows an implementation of a wave function. (Infini-D 3.0 dialog box. © Specular International, Ltd.)

11.2.9 A sequence of animated surface textures where a shiny semi-precious stone turns into carved wood.

still and in a precise location. A more intuitive way to define the position of a camera, however, consists of using the interactive controls to build compound moves.

The three camera moves that are based on a change of the **position of the camera** include a dolly, a truck, and a boom (Fig. 11.3.1a). A **dolly** is a translation of the camera along the horizontal axis. A dolly moving along with a subject, following and tracking it is also called a **traveling shot** (Figs. 11.3.5 and 11.3.6). A **truck** move is a translation of the camera along the depth axis, and is useful for going in or out of the scene. A **boom** is a translation of the camera along the vertical axis. A **crane shot** is a fourth composite move that can be implemented with a combination of position and orientation camera moves. Crane shots with computer-simulated cameras are not bound by many of the physical obstacles—walls, cliffs, trees, fire—that a non-virtual camera has to sort (Figs. 11.3.2 and 11.3.4).

Orientation Camera Moves

The orientation of a camera can be easily defined with an absolute XYZ position value when the camera has a locked point of interest or must be looking in a precise direction. In addition, camera moves defined interactively can be saved as predefined menu items in many computer animation programs.

The camera moves that are based on the change of the **orientation of the camera** include a tilt, a roll, and a pan. A **tilt** is a rotation of the camera on its horizontal axis. A tilt is also called a pivot and is used to look up or down. A **roll** is created by rotating the camera around its Z axis. Roll camera moves are common when simulating fly-throughs. A **pan** is a move created by rotating the camera around its vertical axis (Fig. 11.3.1b). Panning is effective for scanning the scene from side to side while the camera remains stationary. Sometimes, especially when simulating flying cameras, a tilt move is called a pitch—as in airplanes pitching—and a pan move is called a yaw. A zoom is a camera move that is achieved not by moving the position or orientation of the camera but by animating the focal length of its lens (Fig. 7.4.3).

Camera Motion Parallax

The **motion parallax** of a camera describes the visual effect that happens when two objects, one far and one near the camera, move in across the field of vision at the same constant speed. The object nearest to the camera will seem to be moving faster than the object that is far away, even though they are moving at the same speed and traveling the same distance. The reason for this visual illusion is that the object near the camera travels across a smaller area of the field of vision than the distant object has to travel. The object near the camera travels a larger percentage of the field of vision than the object that is farther away (Fig. 11.3.3).

Camera Motion Paths

The technique of motion paths is especially useful for laying out complex compound camera moves—crane shots, underwater shots, and flying cameras in particular—consisting of several individual moves. The motion path technique works by animating an object—the camera in this case—along a path defined in three-dimensional space. Motion paths are similar to the physical tracks used in traditional film production to slide the camera along a predefined path, but without the physical constraints. The paths are drawn with a simple curve modeling tool and edited just as any other object in three-dimensional space would be edited. Motion paths can be created with any type of curve but it is recommended to use b-spline or Bézier curves since both offer superior control for shaping curvature.

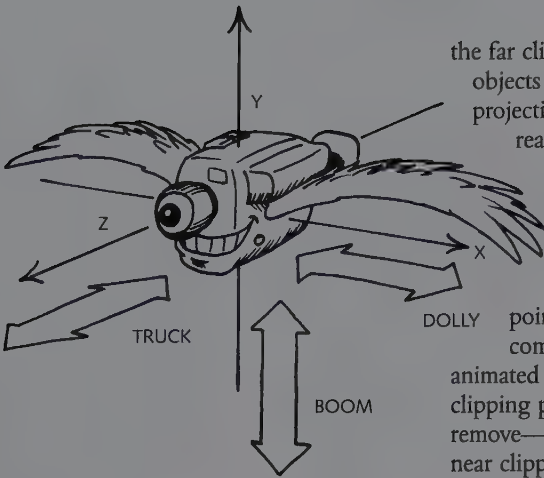
Once a camera is linked to the motion path and once the initial timing parameters of the path are defined, then it is possible to refine the timing, speed, and acceleration with the **animation curves**, another name commonly used for the interpolation graph. Motion path animations can be enhanced with variable speeds, ease ins, and ease outs (Fig. 11.1.7).

The position and orientation of a camera that is animated along a motion path can also be further stylized by adjusting the camera's point of interest, the banking of the camera, or by adding shakes. Many computer animation programs allow you to convert motion paths to explicit transformations. This conversion can help refine the motion by editing the parameter curves of each individual transformation. This simplifies the addition of **camera shake** moves that are typical of **hand-held camera** shots. Adjusting the point of interest of a camera as it moves along the motion path can be used to simulate the **sideways scanning**, or tracking shot, done by characters in a point of view shot as they move down a path (Fig. 11.3.5). The **banking motion** of flying or underwater cameras as they take the curves in a motion path can be defined by linking the camera to an invisible object below or behind it that is subjected to a weight or drag dynamics simulation.

11.2.10 A lot of attitude in this character, Carl, with realistic fur. The NURBS surfaces were modeled with Maya software, and the fur created with a proprietary tool. Many fur characteristics, such as density, color, and clumping, were controlled with image maps painted on the geometry surface. The color along the hairs changes from a dark root to a lighter tip, and the skin color underneath the fur can also be seen. The body and face animation was keyframed with inverse kinematics, and the fur was driven by particle animation and dynamics. (© 2002 Blockbuster Entertainment/Tippett Studio.)

Focal Length and Zoom Camera Moves

The focal length of a camera controls the way in which three-dimensional objects are seen by the camera. The **focal length** in a virtual camera is defined by the relation between the near clipping plane and



11.3.1a The three camera position moves: a dolly, a truck, and a boom. A crane shot can be made with a combination of any of these three moves and also any of the orientation moves illustrated in Figure 11.3.1b.

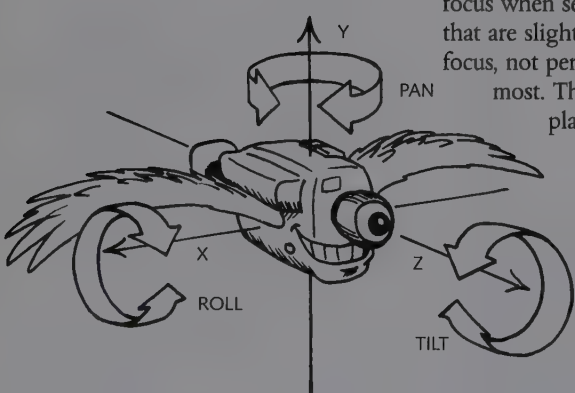
the far clipping plane. This relation defines the way in which the objects in a three-dimensional environment are projected onto the projection plane of a virtual camera—or the surface of the film in a real camera. The focal length in a photographic camera is determined by the curvature and shape of the lens. For that reason, the unit to measure focal length even in virtual cameras is millimeters (mm). Standard camera lenses have a fixed focal length, but **zoom lenses** are capable of variable focal lengths by changing the distance between the point of view and the focal plane (Figs. 7.4.3 and 7.4.5). Some computer animation programs allow for the focal length to be animated independently or in conjunction with the near and the far clipping planes. This provides great flexibility when trying to clip—or remove—an object in the field of vision by placing it ahead of the near clipping plane or beyond the far clipping plane while maintaining a constant focal length.

A **zoom** can be considered a camera move where the camera remains still but the framing of the image changes by gradually and continuously modifying the camera's focal length. In a zoom camera move, both the position and orientation of the camera remain untouched. Zoom moves are ideal when transitions, such as cuts and cross-dissolves, between two shots of the same scene are not desired. For example, going from an extreme close-up to a waist shot, or from a long shot that shows the scenery to a wide shot that focuses on a group of characters.

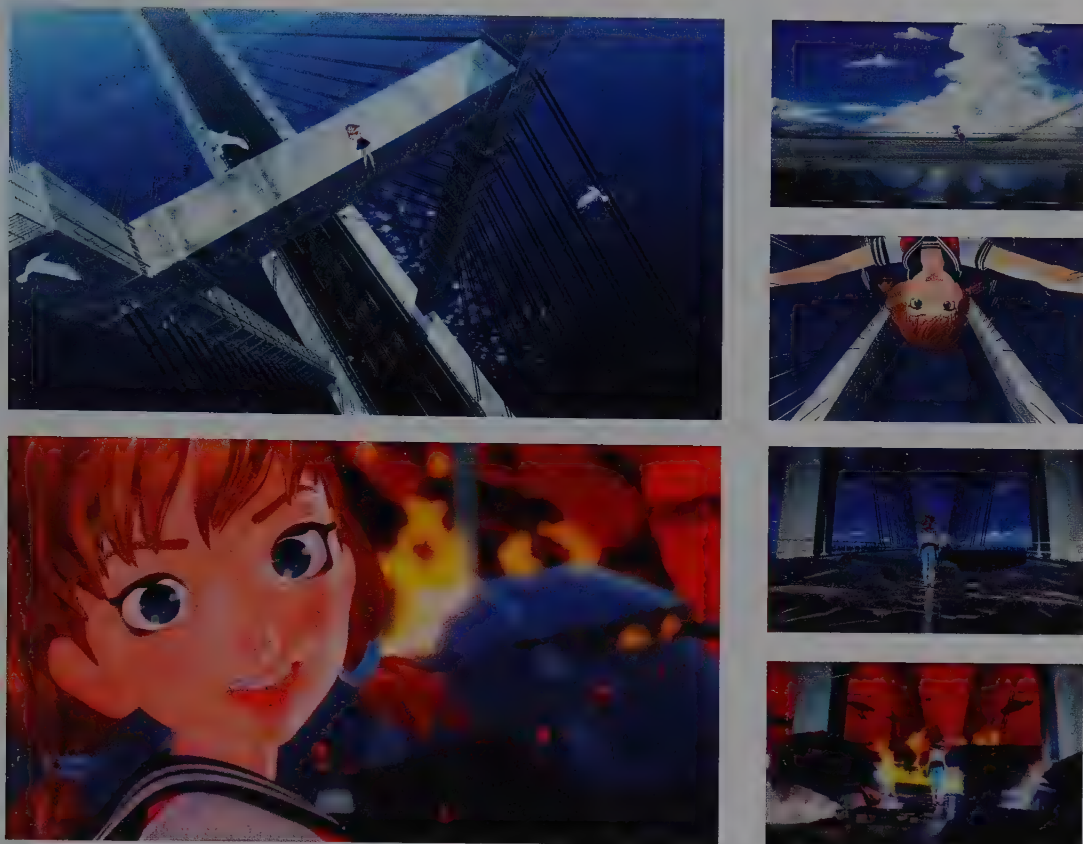
Depth of Field

As mentioned in Chapter 7 the focusing properties of a lens are determined by its depth of field. The focus of a lens defines the plane that is perpendicular to the camera and that will be resolved into a sharp image. Only one plane in three-dimensional space can be in perfect focus when seen through any lens—our eyes included. But the areas that are slightly ahead and slightly behind the focal plane are also in focus, not perfect focus, but close enough so that they look focused to most. The depth of field is bound by the near and the far focal planes, both of which are close enough to the focal plane to still be in focus (Figs. 4.6.3, 7.2.3–7.2.5).

In a real camera, a specific depth of field is determined by the combination of the focal length of the lens used, the **lens aperture** measured in f/stop units, and the distance between the camera and the subject, also called **focal distance**. As a general rule the smaller the lens aperture—which is inversely proportional to the f/stop numerical value—the greater the depth of field. Unlike photographic cameras, virtual cameras are capable of imaging in perfect focus all the objects in a three-dimensional environment. Some programs are capable of controlling focus and depth of field in a way that is similar to traditional



11.3.1b A tilt, a roll, and a pan—camera moves.



11.3.2 Frames from a sequence that is built on the girl's sprint, jump and landing. The camera shows, follows and reveals, using crane shots and traveling shots, but also simple pans and booms. See additional sequence frames in Figure 10.3.1. (*Fly High*. © Keica, Inc.)

cinematography, but many still require an additional pass to **defocus** certain parts or layers of the rendered image. Defocusing layers of a three-dimensional scene is often done in compositing. Using the depth of field and rack focus effects when animating the camera can contribute a realistic flavor to the results. **Rack focus** changes the focus between two focal planes, usually the foreground and a plane behind it.

11.4 Light Animation

The position and attributes of light sources in a computer animation project can be animated using a variety of keyframe and procedural techniques. Some of the former include interactive keyframing, editing of parameter curves, forward kinematics, and motion paths. A wide variety of lighting effects that affect the mood of a scene can be created by animating light attributes such as intensity and color, cone angle, and fall-off. These and other light attributes are described throughout this section and also in Chapter 8.

Writing down the description of the desired light animations in a shot can be a helpful preliminary step before the light is actually

11.3.3 The motion parallax effect has to do with the fact that when a close and distant object move the same distance across the camera, the object closest to the camera (bottom) moves only 25% of the field of view at that point while the distant object (top) moves 50% of the camera's field of view. (Jumping character © 1999 Oddworld Inhabitants, Inc. All rights reserved.)

animated. These written descriptions can provide those involved in the task—cinematographers, lighting technical directors, and animators—a clear idea of what lighting effects are sought and what techniques may be required or are best-suited (Fig. 11.4.1).
A written description of the light animation can be complemented by a simple diagram that annotates the changes in the position or attributes of light in a way similar to a music score. This visual diagram can be a particularly useful reference when scripting the animation of lights with procedural animation techniques.

Moving light sources in a three-dimensional environment should be done with great restraint because moving lights without a clear dramatic or narrative purpose can be a source of visual distraction. Light sources should only be animated when we are trying to achieve a specific change in the mood of a scene. Subtle emotional effects can be achieved, for example, by slowly increasing the intensity of a narrow soft-edged spotlight that is focused on the face of a character. An effect like this one can be especially effective when the **illumination level** in the scene is low. Spotlights that are dimmed or turned up are an effective way of attracting the attention of the audience to a specific area or situation in a three-dimensional scene. Both point lights and spotlights, for example, can be gradually turned up or dimmed throughout an animated sequence by easing their values in or out.

Natural Phenomena

Quite a few **natural phenomena** display dynamic lighting. This type of natural lighting effects include, for example, the light emitted by lightning, fire, or natural explosions such as an erupting volcano; the light reflected off the surface of moving water or refracted through moving water like a waterfall; and the light interrupted by the motion of objects that could be caused by wind in front of a light source (Fig. 11.4.2). The animation of these lighting effects can be done with a variety of techniques depending on whether the light sources were defined with procedural techniques or as a collection of point lights and spotlights. The animation of procedural light sources would simply be done by animating the parameters that were used in the first place to define them (these are reviewed in Chapter 8). Creating lighting effects with procedural techniques can be augmented or replaced with a variety of lighting tricks borrowed from traditional stage and movie special effects. Replicating some of these practical lighting tricks may at first seem crude when compared to the conceptual elegance of motion dynamics simulations, but they are often cheaper to produce than simulations and as effective at impacting audiences.

A trick that is quite common in stage lighting for simulating the light emitted by a small fire or a fireplace consists of using a couple of pulsating spotlights to project light through overlapping strips of colored gel that are constantly waved by a fan (Fig. 11.4.3). The irregular motion of the yellow, red, and orange stripes creates a pat-

11.3.4 Computer and platform games often switch camera positions back and forth between the points of view of the protagonist and an observer. A "third-person" camera follows the action in *Tom Clancy's Splinter Cell Double Agent* with a revealing pan, a tilt and an over-the-shoulder shot. (© 2006 Ubisoft Entertainment. All rights reserved. Splinter Cell, Splinter Cell Double Agent, Sam Fisher, the Soldier Icon, Ubisoft, Ubi.com and the Ubisoft logo are trademarks of Ubisoft Entertainment in the US and/or other countries.)

tern of transmitted light that can be effective when projected on the subjects in the scene. This type of lighting effect, commonly used in both opera and theater, can be easily simulated in computer animation with just a small group of spotlights and point lights and their parameter curves instead of using gel strips, a fan, and spotlights.

Computer animation can also be used to recreate the lighting effects caused by light traveling through falling water, for example; the effect created in an interior space by moonlight, or street lights seen through rain. This trick may include arrays of small three-dimensional models that are animated off-screen between the simulated light source and the scene. In the case of rain, for example, the array of small three-dimensional models could include two or more layers of small cylindrical and translucent shapes that constantly move

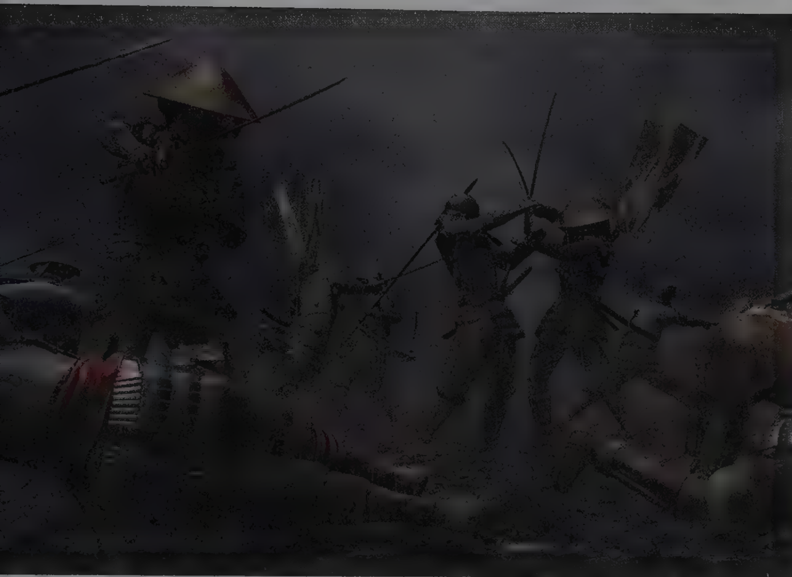
11.3.5 The point of interest is adjusted as the camera moves down the motion path and looks around the scene. (Top: Main character from *The ChubbChubbs.* © 2002 Sony Pictures Imageworks Inc. All rights reserved.)

next to the light source. The two layers of translucent shape patterns can be built in the form of cylinders that rotate around the horizontal axis between the light source and the scene. The resulting top to bottom motions simulates falling water (Fig. 11.4.5). Two or more layers are necessary to avoid a simple pattern of light that is repeated at small intervals. The arrangement of the shapes in each of the layers should also be as irregular and different as possible in order to avoid a repeating motion pattern that is easy to identify. This lighting effect can be maximized by rotating the two cylindrical layers at different variable speeds. The density of the shapes on the rotating layers can yield a variety of effects that range from drizzle to a waterfall. There are alternate versions of this trick—for instance, when the extreme length of a scene may give audiences enough time to recognize the lighting pattern and get bored. One alternative to the rotating cylinders of shapes consists of a long strip with a **translucent image map**—instead of translucent three-dimensional shapes—that is translated from top to bottom between the camera and the scene.

Another variation of the rotating cylinders can be used to simulate the **obstruction of light** caused by objects such as dry leaves being swept in front of the light source. This lighting effect can be achieved by animating groups of flat leaf-shaped models with a pseudo-random factor so that the effect is repeated each time with a slight variation. A primary motion may keep the leaves spiraling in front of the light source, and a secondary motion may keep them rotating around their center or flipping as they rotate. The rotation of several of these groups of leaves can be looped to provide a continuous effect. The effect of the leaves blocking the light can be enhanced by applying a transparency map that makes the leaves transparent on the edges.

The effect of light reflected off the surface of **moving water** can be recreated by placing spotlights shining up through a surface that represents water and that has an animate shape. The lighting effect created by a lightning storm can be recreated by inserting one or two white frames in the sequence just a couple of seconds before the sound of thunder is heard. After that a strong light—placed in the area near where the lightning is supposed to have fallen—is suddenly turned up to a bright white color and then dimmed in a flickering way. The motion of artificial lights during an earthquake, for example, is an interesting convergence of artificial lighting and a natural phenomenon such as those described in Chapter 12.

The light of **celestial bodies**—such as the sun or the moon—usually moves very slowly because those light sources are far from us. An exception to this rule is, of course, shooting stars and comets. The moving light of celestial bodies is usually perceived in the form of **moving shadows** because we can rarely tell that the sun or the moon are moving by just looking at them in real time. (Stop-motion animation techniques can compress real time by recording still frames in a delayed fashion—for example, recording one still frame of the moving sun every minute.) Surreal lighting effects can be achieved by animating celestial bodies at speeds that do not correspond to their



real speeds. The moving light created by a shooting star is a perfect example of a light animation that involves changes in both spatial position based on the speed and distance of the shooting star, and attributes such as the brightness and color determined by the moment when the asteroid enters the atmosphere of the Earth. The light of celestial bodies can be recreated with infinite light sources or with point light sources that have medium to high intensity and little or no fall-off. The color hue of the light of celestial bodies is constant but it may have, for example, a warm tint in the case of the sun or a slightly cold tint in the case of the moon. Short of procedural or parametric techniques, the flickering effect of shooting stars can be implemented by creating an irregular pattern in the parameter curves that control the animation of the color or the fall-off of the light source.

Practical and Artificial Lights

Many night or interior scenes have **practical lights** that move or change during the shot. Practical lights are reviewed in Chapter 8 and include flashlights, matches, torches, table lamps, and even fireflies. Many of us have been fascinated by the blinking patterns of light created by fireflies as they fly through the night. Other animals—such as the fluorescent fish that live in the depths of the oceans—are also light sources that move naturally. The light emitted by fireflies can be simulated with point lights or with spotlights that have a wide cone angle and a narrow spread angle. Firefly light has a large amount of fall-off because it does not travel far, and its color can be animated within a narrow range of fluorescent green hues. The blinking pattern of firefly light can be replicated by drawing parameter curves for color, fall-off, or cone angle so that they are

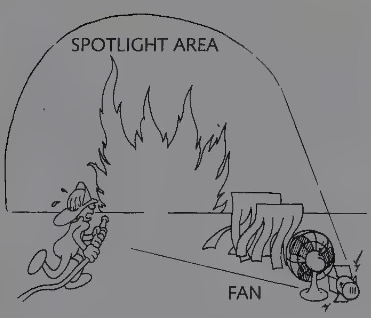
Written Description of a Light Animation Sequence

- Shot length is 10 seconds, and starts in darkness.
- Takes place in a room with a a window, and a round table with a wooden sculpture of a bird on it.
- We hear the crackling sound of fire outside, but the flames are barely visible through the half-closed window shades.
- Suddenly a strong wind current throws the window open; a whirling point light flies into the room, and glows more intensely as it circles over the table.
- The light sparkles and hovers, floating up and down.
- Sparks land on the table and the bird; the latter glows gently for a couple of seconds, it flaps its wings and flies out the window.

11.4.1 Written descriptions of a light animation can help to clarify what techniques may be required to produce the sequence.

11.3.6 (Top left) Cinematic sequence for the *Onimusha* game, awarded Best of Show at SIGGRAPH 2000. (© 2001 CAPCOM Co., Ltd. All rights reserved. Guest creator: Takeshi Kaneshiro. CGI by Links Digiworks. Composition © Mamoru Samuragoch. Character Samanosuke Akechi © Amuse/Fu Long Production.)

11.4.2 An early example of natural phenomena lighting. (Courtesy of Rhythm & Hues Studios.)



11.4.3 Fire effects can be simulated by projecting light through strips of colored gel moved by a fan.

11.4.4 Parameter curve of the light emitted by a firefly.

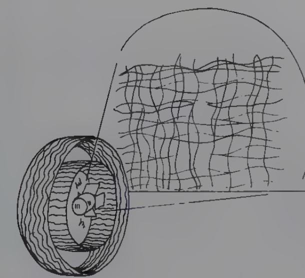
interrupted with abrupt jumps. The parameter curve in Figure 11.4.4 suggests the abrupt jumps in cone angle values that represent a blinking light. The small zig-zag variations along the vertical axis represent flickering, and the abrupt jumps and sharp 90-degree angle changes in direction represent blinking. The flat horizontal lines represent a constant darkness achieved by a cone angle value of zero.

Theatrical or **artificial lights** can be animated as point lights or spotlights. Artificial still point lights include, for example, a bare lightbulb. Man-made moving spotlights include, for example, the light reflectors used in stage or movie productions—and so are commonly associated with Hollywood—the light projected in darkness by moving vehicles, or the light projected by flashlights or other appliances such as open refrigerators, copy machines, and televisions that are turned on in darkened environments. Realistic simulations of artificial lighting should pay attention to the color temperature of the light source. Animated spotlights in a dark scene can add a feeling of suspense or fear to the shot because the lighting effect may remind the audience of a search for something—or someone—that is hiding, or trying to hide from someone—or something—that is searching. Evening lights in the urban landscape are commonly used to set the mood in an establishing shot (Fig. 11.4.6).

11.5 Hierarchical Character Animation

Three-dimensional objects can be grouped together into **hierarchical structures** that define the ways in which these objects relate to one another and behave when animated. The basic idea behind a hierarchical structure in computer animation is that it controls the order in which transformations are applied to the joints in the rig that animates the geometry. (See Chapter 5 for additional features of hierarchical structures as applied to the rigging of models.)

Diagrams of hierarchical structures are often represented as an inverted **tree structure** where the highest level of importance in the structure corresponds to the trunk of the tree. The main **branches** that come directly out of the trunk represent the next level in the hierarchy; branches that come out of the main branches are at the next level and so on, until we get to the leaves, which represent the last level in the hierarchical structure. The objects within the hierarchy inherit attributes—including motion—from the dominant objects just as children inherit attributes from their parents. It is also possible to animate just a selected branch in the structure without having to animate the entire structure. The relationships between objects—or parts of objects—in a hierarchical structure can be easily visualized with a schematic representation of a **hierarchy diagram**. These line diagrams often consist of boxes that represent the items in the structure, and lines that represent the place of the items in the hierarchy and their relations with other items. In most cases, there is just one set of hierarchy diagrams per scene, and these diagrams control the animation of all the objects.



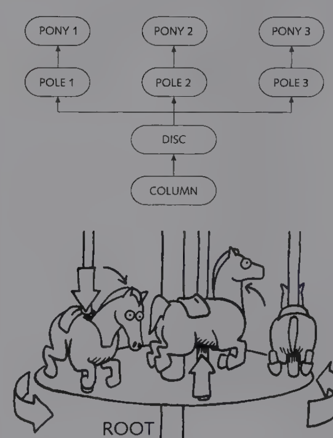
11.4.5 The effect of light being projected through rain or a waterfall can be achieved by rotating two cylindrical layers of translucent shapes at variable speeds between the light source and the scene.

11.4.6 This nocturnal scene from the *Blade Runner* computer game displays a plethora of artificial lights traveling through the fog and casting shadows and interreflections. (© Westwood Studios. All rights reserved.)

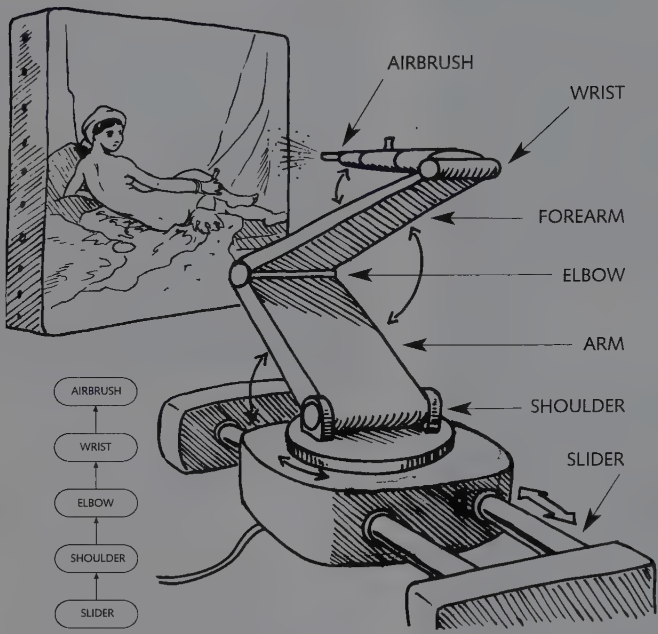
Levels of Precedence

Objects within hierarchical structures have well-defined **levels of precedence** or importance. The object or objects at the top of the hierarchy are called **parents**, and the objects below are called **children** and **grandchildren**. The most dominant object in a hierarchy is usually called the **root** of the hierarchy, and objects that are placed in the same branch of the hierarchy or at the same level in the hierarchy are often called **siblings**. A **null parent** is a node in the hierarchy that does not relate to any specific part in the model but that controls several child objects together. A null parent is used, for example, when two or more objects are grouped at the same level in the hierarchy. Nulls are usually represented as empty boxes in the structural diagrams. Figure 5.7.1 illustrates the use of a null node.

Hierarchical structures sometimes include objects that are assembled into an articulated figure. They may also include objects that are not physically connected to one another. **Articulated figures** with hierarchical groupings of objects are an essential tool for the creation of computer-based character animation. There are many possible ways to group several objects in a hierarchical structure, but the hierarchy of parts in a model should always be driven by the motion requirements. Figures 11.5.1 and 11.5.2 show an articulated and a nonarticulated model, each with their corresponding hierarchical diagrams. The hierarchy in both examples is quite simple since there is minimal branching.



11.5.1 A merry-go-round is a good example of a hierarchical structure with multiple levels, where the motion of the objects in the parent levels determines the motion of those objects in the children levels. Everything rotates together, but each horse has a unique vertical motion and timing.



11.5.2 This spray-paint robot arm is an articulated figure with three joints. The shoulder has two degrees of freedom, and the elbow and wrist each have only one degree of freedom. The diagram on the left represents the hierarchical structure with the root placed at the slider

Joints and Degrees of Freedom

The type of **joints** used in computer animation are defined by the number of degrees of freedom that they have. **Degrees of freedom** express the ability of a joint to rotate around and/or to translate along one or several axes (Figs. 11.5.3 and 11.5.4). One degree of freedom, for example, corresponds to the ability of a joint to rotate around one axis, while a joint with three degrees of freedom is capable of rotating around three axes. The knee, for example, is a joint with only one degree of freedom, whereas the shoulder has three degrees of freedom (go ahead, try it, you can rotate your arms around the X, Y, and Z axes). Joints can be catalogued according to their degrees of freedom from a simple one-dimensional **twist joint** or **bend joint** to a multidimensional **universal joint** that can rotate in all directions (Fig. 11.5.5). In addition to the number of degrees of freedom, joints are also defined by a **rotation range**, which restricts the rotation of the joint between a minimum and a maximum value. A joint can have a different rotation range for each of its degrees of freedom. These **motion constraints** imposed by rotation ranges are essential when animating articulated figures with the inverse kinematics techniques described in Chapter 12.

The centers of objects—often called **centroids** or pivot points—play an important role in the hierarchical animation process because many operations are calculated based on their position. These operations include all the geometric transformations as well as simulations of motion dynamics related to the center of gravity. By default, most three-dimensional programs place the **pivot points** in the geometric center of the objects but they can be interactively repositioned anywhere in the object.

Software that supports hierarchical structures offers a hierarchy editor, also called a **skeleton editor**, to create links between objects and to set **joint information** such as stiffness and **range of rotation**. Skeleton editors are usually based on a graphical diagram that shows the objects in the connected hierarchy and dialog boxes with information for each item in the diagram. **Links** between objects can be established by clicking directly on the objects. Another method establishes hierarchical relationships by clicking on the boxes in the diagram that represents the links between the three-dimensional objects (Figs. 5.7.2 and 12.1.9). Most programs that build the hierarchy from top to bottom require that the parent object is clicked before the children.

11.5.4 These titanic characters from the *Command and Conquer 2* computer game have different types of joints and degrees of freedom. Can you figure out what they might be? (© Westwood Studios. All rights reserved.)

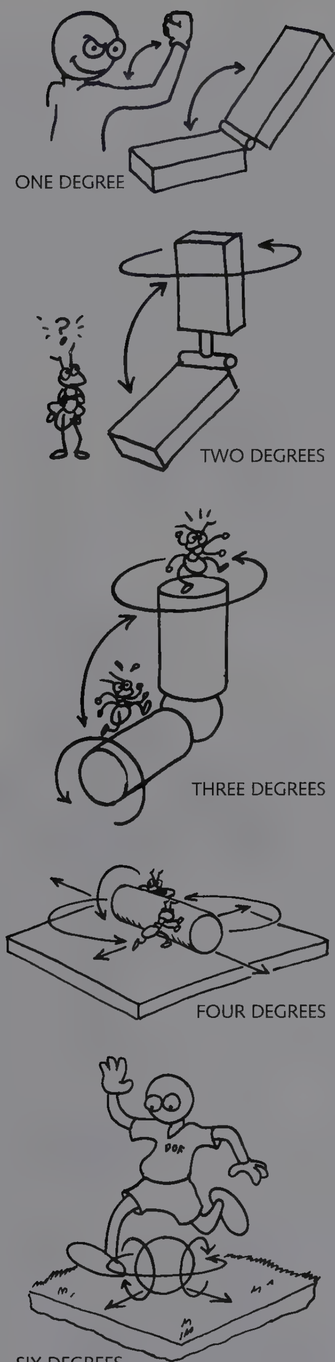
11.6 Two- and Three-Dimensional Integration

Combining and matching two-dimensional hand-drawn animation with three-dimensional computer renderings was pioneered in animated works like Disney's *The Great Mouse Detective* (1986) and *Beauty and the Beast* (1991), or *Technological Threat* by Bill Kroyer (1988, Fig. 1.3.10). At the time, this technique represented a major challenge, but integrating two-dimensional and three-dimensional elements in the same shot is today a standard technique in the computer animation toolbox. Many animated movies and TV series that are primarily hand-drawn incorporate a fair amount of significant three-dimensional elements: the crowds in *Prince of Egypt* (Fig. 12.6.2), the octopus-tarantula monster in *Princess Mononoke*, and the space vehicles in Matt Groening's *Futurama* and Disney's *Atlantis* and *Lilo and Stitch*. "Tradigital" is the clever name that DreamWorks' Jeffrey Katzenberg has given to this melding of styles.

The technique of **two- and three-dimensional integration** is also known as 2D/3D integration, and it has a few variations, each suited for a different style. These variations include two-dimensional hand-drawn characters on three-dimensional computer-generated backgrounds, three-dimensional computer-generated characters and props on two-dimensional hand-drawn backgrounds, and stop motion animation characters on three-dimensional computer-generated backgrounds. Combining these styles with live action is also possible (Fig. 11.6.2). Each 2D/3D integration style requires a slightly different process but compositing is always an essential component.

The process for integrating two-dimensional hand-drawn characters on three-dimensional computer-generated backgrounds starts by laying out the shots on the three-dimensional software. This layout

11.5.3 This *TMNT* character is animated with a hierarchical rig and the muscle system built over it. The muscle system is used to deform and shape the character's surface geometry. (Teenage Mutant Ninja Turtles and TMNT are trademarks and copyrights of Mirage Studios, Inc. *TMNT* © 2007 Imagi Production Limited.)



11.5.5 A variety of joint types with different degrees of freedom.

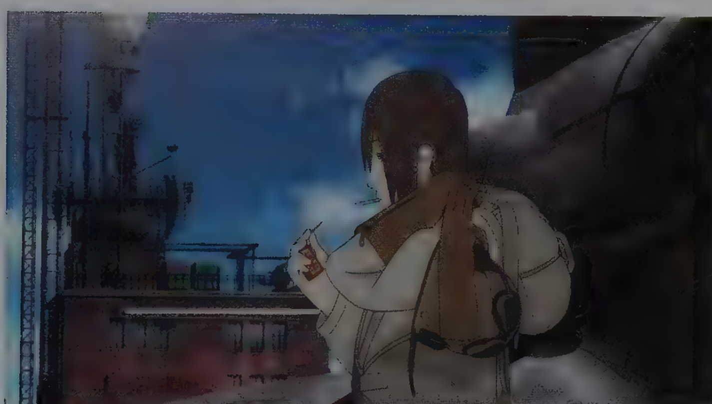
takes into account the action of the characters regardless of whether the shot involves camera moves. Each keyframe, usually in wireframe mode, is used by the two-dimensional animators as templates for blocking and animating the characters. Computer-rendered backgrounds and handmade drawings are usually created at the same field size (Fig. 7.2.2). In shots with a moving three-dimensional camera, animators use some or all of the continuous frames of the background to match their drawings to the speed and perspective of the camera. The progress of the two-dimensional animation can be checked or previsualized by quickly compositing the drawings with the three-dimensional backgrounds. Drawings can be created with pencil and paper or digitally with a graphics tablet. Once the two-dimensional animation is completed and approved the drawings can be scanned in high resolution, inked and colored digitally, and composited in batch mode with the three-dimensional backgrounds (Figs. 6.11.1, 11.6.1 and 11.6.3). Shots with extreme camera moves or complex action might require 2D or 3D rotoscoping techniques to match the two-dimensional hand-drawn animated characters to the three-dimensional backgrounds or virtual sets (Fig. 13.3.1).

In the case of three-dimensional characters animated on two-dimensional backgrounds the integration process is usually simple. The two-dimensional background is scanned and used as a template to adjust the position and orientation of the three-dimensional camera (Fig. 12.6.2). This can be done by matching the position of the camera visually by trial and error, positioning a three-dimensional grid overlay on the two-dimensional painting. This can also be accomplished by using camera tracking software to aid matching the perspective of the three-dimensional camera to the perspective of the two-dimensional background (Fig. 13.2.1). It is also possible to model both the environments and all or some of the models in three dimensions, use the three-dimensional environment as a template to paint two-dimensional backgrounds, and composite the painted backgrounds with the three-dimensional characters (Fig. 6.11.1).

Combining characters created with stop motion animation techniques and three-dimensional computer-generated backgrounds provides another opportunity for two- and three-dimensional integration. In this case the integration process is fairly similar to the process of camera tracking for live action footage described in Chapter 13. The position of the virtual camera is calculated with camera tracking software by providing the position and orientation of the camera used to record the stop motion as well as tracking markers used on the stop motion set (Fig. 11.6.4).

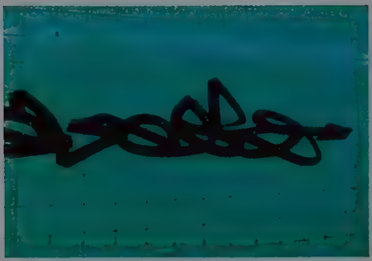
11.7 Animation File Formats

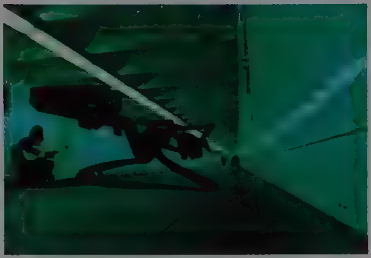
Three-dimensional computer animations can be saved in a variety of file formats after they have been rendered. Those output file formats are described in detail in Chapter 15, and include QuickTime, MPEG, and AVI formats. However, before the animations are rendered they



11.6.1 *Sky Crawlers* features two-dimensional animation composited with After Effects on three-dimensional computer renderings with matching lighting and perspective. Props and sets were modeled and animated with Maya, clouds with Vue software. Mental Ray shaders used include normal map, ambient occlusion and HDR textures, and motion blur was used to render the airplane propellers. (Images courtesy of Polygon Pictures. © MH/NI, BWDVYHDYCH.)

usually exist in proprietary formats that can only be processed by the application program that created them. Most computer animation programs are able to save the animation parameters and data in a **standalone** animation file that is independent from the files containing the modeling and rendering information. However, unlike many modeling and rendering programs capable of saving data in both native and portable file formats, most animation programs only save the animation information in native file formats that are incompatible with other programs. The **FBX** file interchange format is an exception to this rule. Developed by Kaydara, FBX is a widely supported format used for acquiring and exchanging three-dimensional assets and media from a wide variety of sources. XYZ translation motion data captured with an optical system can be saved in a variety of formats





11.6.2 Animated graffiti comes alive from two to three dimensions in a New York City subway station. (Images courtesy of Psyop, Inc.)

11.6.3 (Facing page, left) Street scenes in La Habana, from the movie *Chico and Rita*, rendered by projecting hand-drawn illustrations onto geometry, including billboards for people on the street. (Facing page, right) Frames showing the style of the two-dimensional animation that was integrated with some three-dimensional environments. (Images courtesy of Fernando Trueba P.C. S.A. and Estudio Mariscal S.A., U.T.E., right-owners of the film *Chico and Rita*.)

including Acclaim (ASF and AMC) and Biovision (BVA and BVH). These formats provide the post-processed data once it has been applied to the skeleton. The Acclaim format—adopted by the Vicon mocap system—is saved in two related files, an **ASF** file with the skeleton hierarchy and an **AMC** file with the motion capture data. The **BVA** format includes nine channels of positional data with XYZ translation, rotation, and scaling values. The **BVH** format saves the hierarchy and initial pose of the skeleton, and the motion data.

11.8 Getting Ready

Choose Appropriate Motion

Audiences learn a lot about the emotions and intentions of animated characters by the way they move. Make sure that the motion applied to the models matches the dramatic purpose of the scene and its level of rendering realism (Fig. 11.8.1). Realistic motion for the models might be necessary for realistic renderings, while limited motion may match a simple rendering style.

Avoid Still First Frames

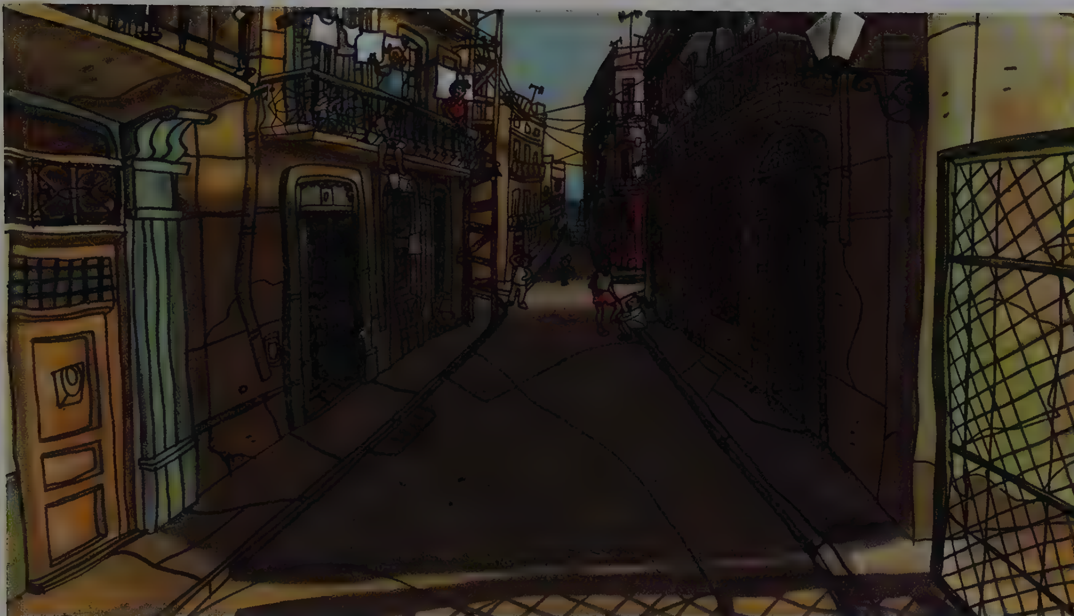
One or several still frames at the beginning of an animated sequence usually look like a mistake, as if when the camera starts rolling before the actors are ready. Unless the sequence calls for several frames of a motion hold, avoid starting your animations with models that are still. You can improve the sense of continuity and motion flow by starting the scenes with characters that are in mid-motion, regardless of how subtle that motion may be.

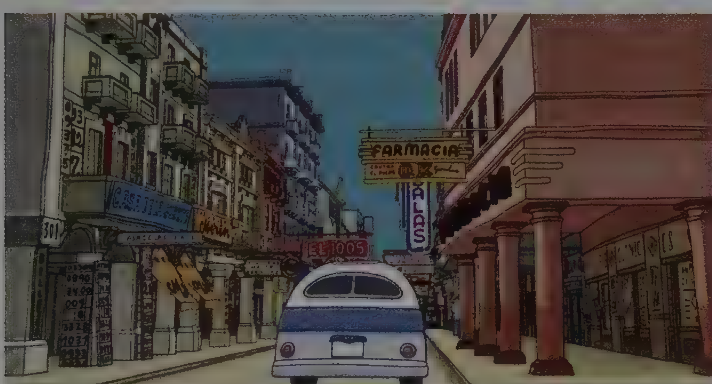
Preview the Motion

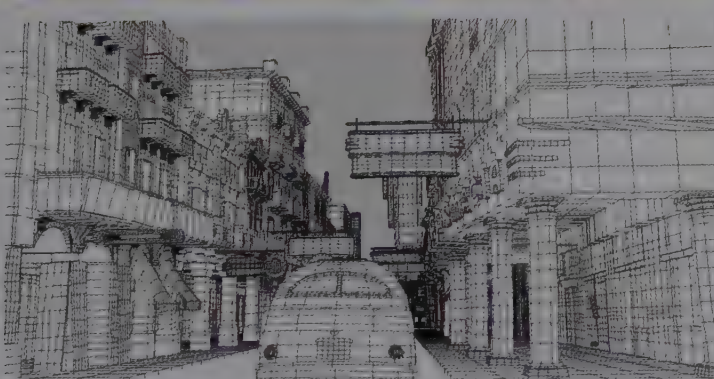
Virtually all computer animation programs allow for the creation of motion flipbooks that can be played back directly onto the computer screen. A **digital flipbook**, also called a **playblast**, consists of a sequence of image files displayed in an area of the screen, ranging from a small window to the full screen. Digital flipbooks can be played back at their final output speed—for example, 24 or 30 frames per second—or at slower speeds to analyze the animation in more detail. Before digital flipbooks were available, motion tests could only be previewed by recording each frame onto a videotape, and playing them back on a videotape player. Those days are fortunately gone.

Check the Preferences File

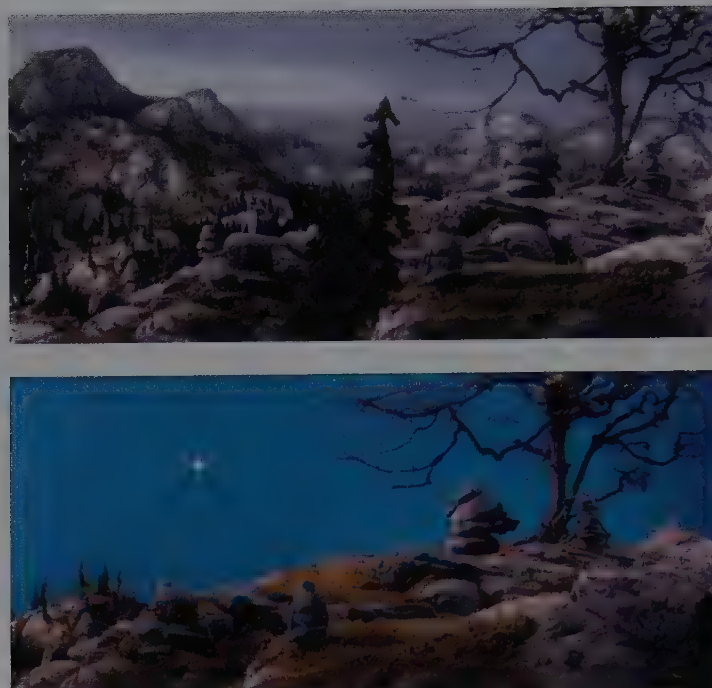
The default settings stored in the preferences file or dialog box are important because they control directly and indirectly the result of many operations, functions, and tools of three-dimensional animation programs. Some of these settings include, for example, the ani-




FARMACIA
SALAS



11.6.4 A finished frame from *Das Rad* composed of a three-dimensional computer-generated background and a stop-motion foreground. The latter can also be seen against a blue screen, with a white marker to facilitate camera tracking. (© 2001 Filmakademie Baden-Wuerttemberg/Gruber/Stenner/Wittlinger.)



11.8.1 (Opposite page) User-controlled dynamic cameras allow players to navigate through a computer game. (Images courtesy of Naughty Dog, Inc. Naughty Dog and the Naughty Dog Logo are registered trademarks of Naughty Dog, Inc., a wholly owned subsidiary of Sony Computer Entertainment America, Inc. *Jak & Daxter* is a trademark of Sony Computer Entertainment America, Inc. Created and developed by Naughty Dog, Inc. © 2001 Sony Computer Entertainment America, Inc. *Jak & Daxter* was developed for the PlayStation 2.)

mation rate of frames per second, the aspect ratio of the still image, and whether to load external files—such as texture maps or custom procedures—that may affect the final look of the animation project.

Preview with Multiple Camera Views

It is helpful to use multiple active camera views when setting the keyframes in a three-dimensional animation. The camera perspective view is useful for previewing the motion from a specific point of view. But the other camera views, including the front view, side view, and top view, are useful for checking details, such as overlapping objects.

Follow the Animatic or Storyboard

When animating it is important to stick to the directions specified in the animatic or storyboard because other individuals may be working at the same time on the same sequence as you are. If you decide to interpret the storyboard liberally you may create animated sequences that might not match the previous shot or the one that follows, or that will look different from shots of the same sequence done by others. Consult the animation supervisor when you have an idea for improving the portion of the storyboard that you are working on. Maybe your idea can be carried out but only after other team members have been consulted. Some productions can afford this type of experimentation while others cannot.

Chapter 11

Key Terms

Absolute position
Acceleration
Animation curves
Articulated figures
Artificial lights
Attributes
Banking motion
Bend joint
Boom
Branches
Camera position, orientation
Camera shake
Celestial bodies
Centroids
Children
Circular waves
Constant speed
Crane shot
Curved interpolation
Defocus
Degrees of freedom
Digital flipbook
Dolly
Ease functions
Ease in, out
External control structures
Fantastic transformation
Focal distance
Focal length
Forward kinematics
Free-form lattice
Free-form shape animation
Function curve
Grandchildren
Hand-held camera
Hierarchical structures
Hierarchy diagram
Illumination level
In-between frames
Interactive keyframing
Interpolation ease
Joint information
Joints
Key poses
Keyframe
Keyframe interpolation
Lens aperture
Levels of precedence
Linear interpolation
Linked to path
Links
Locked shots
Mapping a sequence
Motion constraints
Motion parallax
Motion paths
Moving shadows
Moving water
Natural phenomena
Natural transformation
Null parent
Obstruction of light
Order of correspondence
Pan
Parameter curve
Parents
Pivot points
Planar waves
Playblast
Practical lights
Rack focus
Range of rotation
Rate of change
Roll
Root
Rotation range
Shape animation
Siblings
Sideways scanning
Skeleton editor
Slope
Spatial animation
Speed
Spherical waves
Squashing
Stretching
Three-dimensional morphing
Tilt
Timeline
Timing of transformations
Transformation matrix
Translucent image map
Traveling shot
Tree structure
Two- and three-dimensional integration
Truck
Twist joint
Universal joint
Wave functions
Zoom
Zoom lenses

CHAPTER 12

Advanced Computer Animation Techniques

QUICK INDEX

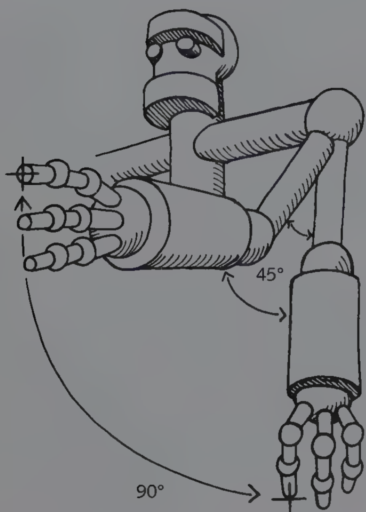
Summary

MOST OF THE ADVANCED ANIMATION TECHNIQUES covered in this chapter are quite different from those techniques based on the traditional keyframe approach. These techniques are used to simulate complex or realistic motion of objects and characters. Many of these techniques, in fact, start by capturing the motion of real actors and applying it to animated characters. This chapter also presents the **hybrid environment** in which some of the latest advanced animation techniques are almost always used in combination with others. The concept of working in layers or channels of motion is stressed throughout the chapter. One of the main reasons for using hybrid animation techniques is the fact that natural motion is too complex to be recreated with just one technique. For example, the motion of three-dimensional models can be controlled in detail if we provide their positions and angles to an inverse kinematics program, but their motions may not be physically correct. Likewise the motion of models will be realistic if their motion dynamics are simulated based on the forces applied to them, but it will be difficult to obtain a specific motion especially as the models become more complex.

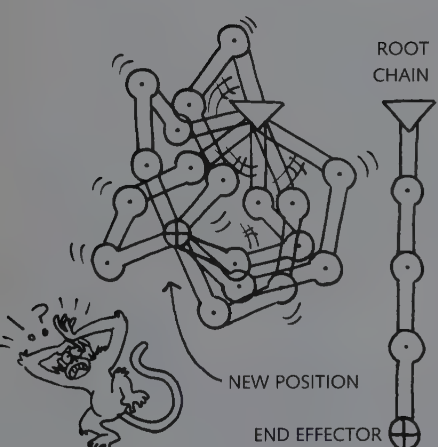
12.1 Inverse Kinematics

Inverse kinematics, or **IK**, techniques are useful for animating complex models and motion rigs with a large number of joints. Unlike forward kinematics, their brute force counterpart, **inverse kinematics** techniques determine the motion of the skeleton of a character based on the final angles of some of the key joints that define the motion. Inverse kinematics requires that the three-dimensional models to be animated are built as hierarchical structures. Inverse kinematics techniques are most commonly applied to articulated figures that are defined as hierarchical skeletons of rigid parts connected by joints, each with its own motion constraints. **Hierarchical skeletons** are composed of many articulated chains grouped together in a hierarchy.

12.1.1 (Opposite page) Woody and Buzz Lightyear, the unforgettable characters from *Toy Story*, the first animated feature film produced entirely with three-dimensional computer animation techniques. (Images courtesy of The Walt Disney Company. © Disney Enterprises, Inc. All rights reserved.) (Above: Image courtesy of Framestore/Monster.)



12.1.2 The end effector of an articulated chain representing an arm is usually the hand or a fingertip.



12.1.3 An inverse kinematics sequence that starts (A) without motion or position constraints may result in endless motion variations when the end effector is repositioned (B of Figure 12.1.6, opposite page).

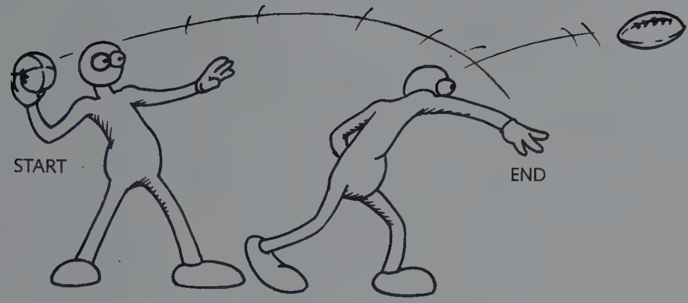
Skeletons are used to control the deformations of the surfaces surrounding them as well as the motion of additional geometries that might be attached to them.

Inverse kinematics techniques can greatly simplify the work involved in animating characters, especially those that move in a complex and realistic way. Trying to animate a running tiger, for example, with just forward kinematics could turn into a long and time-consuming process of trial and error, especially if the tiger was running on an uneven terrain that had obstacles scattered along the way. But the same task can be simplified by using inverse kinematics because this animation technique can use the position of the paws' last joint, for example, to animate the entire figure into the desired positions. The components of the inverse kinematics technique include a hierarchical rig, joints, motion constraints, and effectors. Articulated figures with hierarchical chains allow simultaneous movement of all their parts but always following the specified hierarchy.

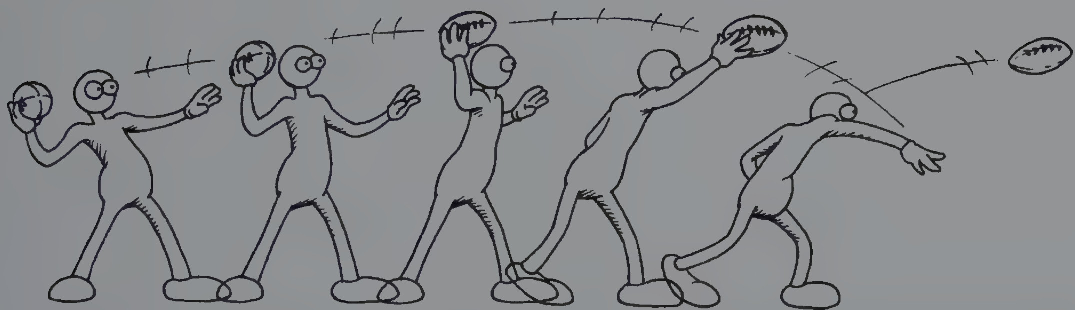
An articulated hierarchical chain is composed of a chain root, a certain number of joints, and an effector. The **chain root** is usually the first joint in the first segment of a simple articulated chain. The chain root is often the parent of all the segments and joints in the articulated chain. The **effector**—also called **end effector**—in a hierarchical chain is the joint that is used to determine the positions of a moving chain with inverse kinematics. When the effector in a chain is moved, the inverse kinematics are invoked and the rotations of the joints are automatically calculated. In the case of an arm reaching to push a button, for example, the end effector of the motion would be located in the hand or an extended fingertip (Fig. 12.1.2). The end effector of most arm motions is, in fact, usually placed in the hand or the fingertips.A **joint** is defined by the articulation point where two segments of the articulated chain meet. Most inverse kinematics programs allow the joints in the chain to be rotated in any direction unless motion constraints have been placed on the joint. A hierarchical chain can follow the motion of an end effector in many different ways whenever motion constraints are not set (Fig. 12.1.3).

Animating complex articulated figures with inverse kinematics is advantageous because if the figure has the proper joint motion constraints, the motion of a single end effector is used to determine how all the joints in the figure must rotate. The entire figure follows the motion of the end effector. If the joint **motion constraints** have been set in a logical way that leads to the desired motions, then inverse kinematics techniques can save work during the animation process. In some cases, forward kinematics offers more direct and immediate control of the joint positions at any point during the animation process (Fig. 12.1.5). But in most situations, animating a complex articulated figure with inverse kinematics is often more efficient than using forward kinematics (Fig. 12.1.4).

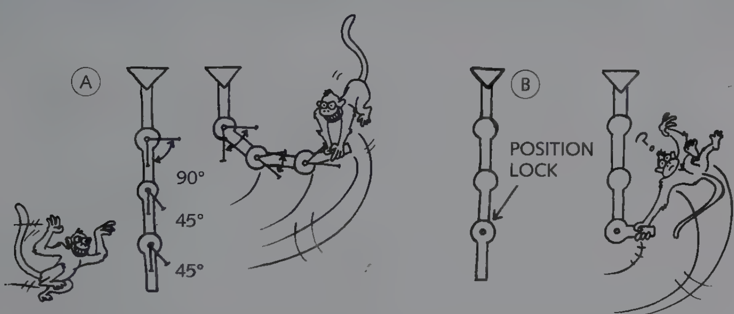
Assigning motion constraints to each joint is necessary to limit the results when the end effector of the chain is dragged to a new



12.1.4 Throwing a ball with inverse kinematics. Only the starting and ending positions need to be specified.



12.1.5 Throwing a ball with forward kinematics. All joint angles need to be specified.



12.1.6 Three inverse kinematics sequences of a simple hierarchical chain. They all follow the same end effector but each has different motion constraints including multiple rotation constraints (A), and a single position constraint (B).

position (Fig. 12.1.6). Animating a hierarchical chain with the same end effector and no motion constraints can generate many different results (Fig. 12.1.3). Motion constraints are often expressed in terms of degrees of freedom and rotation angles.

The hierarchy in the skeleton of a complex articulated figure can be broken to facilitate the animation of a part or limb that should not follow the motion of the chain root. That would be the case, for example, of a character whose feet must remain on the ground or whose hands must remain attached to an object even as the chain root—usually located somewhere in the hips or torso—moves. During a **broken hierarchy**, some or all of the limbs are not controlled by the chain root but instead have their own root. In a broken hierarchy, a hand, for example, is not directly controlled by the torso through the shoulder (Fig. 12.1.7).

Inverse kinematics is an effective technique for laying out actions by specifying key poses at each keyframe, first primary

12.1.7 The pivot points in the skeleton that controls a hand animation are shown as diamonds (top). The yellow diamonds and webbing between the thumb and the index are flexible links that behave like springs. A red sphere on the palm (top) is a pivot point that connects the hierarchical root of the skin to the skeleton. The wrinkles in the knuckles are modeled on the mesh (middle), and a material ID number has been applied to the fingernail mesh that is different from the skin material. (© 1999 Mondo Media, San Francisco, California.)

motions and then secondary motions. When animating the hierarchical model of a human-like body (Figs. 12.1.8 and 12.1.9) it is usually best to animate down the hierarchy, starting with broad motions first and details second, animating from the inside to the outside, the torso before the head or arms. This approach can be complemented with the techniques presented in the next section of this chapter.

12.2 Performance Animation and Motion Capture

Real-time **motion capture** is an advanced animation technique that allows animators to capture live motion with the aid of a machine and then apply it to computer animated characters. Motion capture, also known as **mocap**, is different from traditional keyframe animation because it captures all the motions of live actors as they move. Motion capture can also be used to create the **basic tracks of motion** that can later be enriched with other animation techniques. Much of the **secondary motion** in an animation based on motion capture techniques—such as the detailed animation of hands, fingers, and facial expressions—is usually added on top of the basic tracks of primary motion (Fig. 12.2.3).

Motion data that is captured and saved as XYZ joint positions can be manipulated directly and also applied, for example, to an inverse kinematics skeleton. Some motion capture methods are better suited for **live control** of animated characters, while others are more adequate for situations that require complex motion sequences with **multiple layers of motion**. Except for when the motion capture data is used to control live motion, the data collected with motion capture systems is subjected to different amounts of clean up and refinement in the computer animation system. This is because the **raw motion** captured often contains too much noise that has to be cleaned out or because it is not sufficient to generate by itself the motion required in the animation sequence.

A major attraction of motion capture techniques is that they may be used to produce animation in a cost-effective way, but only after all the initial setup has been worked out. Depending on the nature of the project, motion capture techniques make it possible to automate large portions of the character motion, and they can also eliminate some of the keyframe-based manual work.

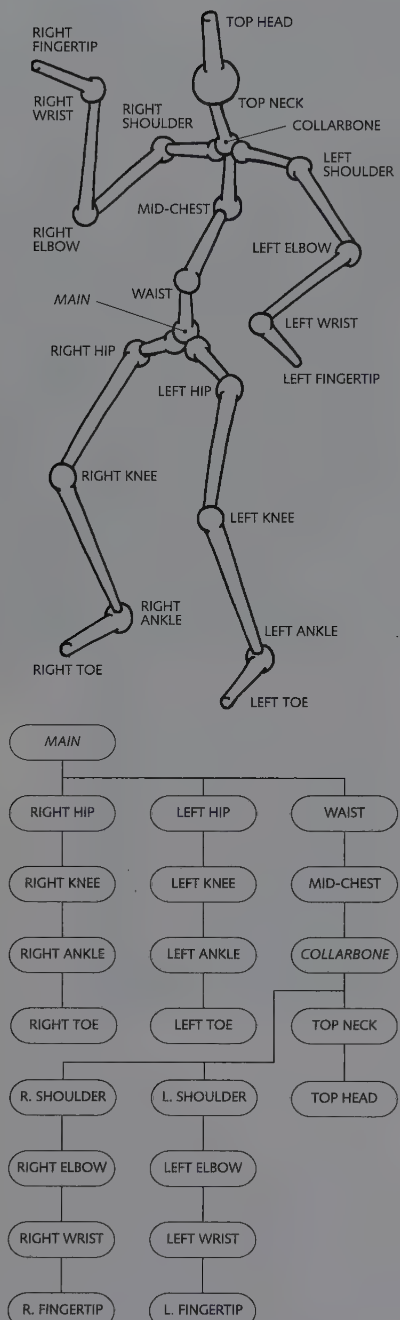
Motion capture techniques began to be practical in the early 1980s when researchers experimented with potentiometers attached to human bodies for measuring the angles of joints. Light emitting diodes (LEDs) and mechanical armatures were also used early on to measure both the position and orientation of joints. Most of the early applications of motion control were limited to animating live simple cartoon characters or heads and faces, but not full body animation. These live animations would often be composited with live action. In many cases, a variety of helmets and armatures have been used to capture the motion of the face and body of an actor. Today, even

though motion capture animation techniques are still being refined, there is an increasing variety of character animation based on motion capture techniques. Many off-the-shelf computer animation packages offer hookups for a range of motion capture gear that can bring motion data directly into the animation system. The prejudice about motion control being a technique that is used only by those who do not know how to animate the "old-fashioned way"—entirely by hand—is also slowly giving way to more progressive points of view.

Motion capture implies one or several real actors that generate motions for one or several animated characters (Fig. 12.2.2). Preparing both the **real actors** and the animated characters—or **virtual actors**—for the process of motion capture involves two somewhat independent series of steps: setting up the capture points on the human actor and setting up the hierarchical structures that will control the virtual actor. The exact positioning of the sampling points depends on the type of motion desired. But in all cases, it is necessary to establish a correspondence between the **sampling points** in real actors and the joints in the animated characters. Illogical correspondences between sampling points and joints in the animated characters can lead to unexpected and amusing results. Imagine the motion of an animated character whose neck joints are animated by the motion collected by the sampling points in its tail!

Motion capture is often used to capture primary motion, so sampling points are often distributed throughout the head, torso, and limbs. Secondary motions such as facial expressions and hand gestures

12.1.8 Poses and actions are best animated from the inside out, starting with the broad motions of shoulders, torso and hips, continuing with the head, arms and legs, and finishing with facial expressions, hands, feet, hair and tail. (*Madagascar: Escape 2 Africa*™ and © 2008 DreamWorks Animation LLC, used with permission.)



12.1.9 A simple articulated model and its corresponding hierarchical diagram.

are often added with other animation techniques to the captured primary motions. It is also important to make sure that the hierarchical structures of both real and virtual actors are structured so that the captured motion will result in the desired effects. Computer animation projects have a wide range of requirements in terms of the minimum number of both sampling points and joints in the figure. It is not necessary that the hierarchical structures be identical, but when the hierarchies of real and virtual actors are structured in different ways, the resulting motion will not be a direct translation of the captured motion. In these cases, the resulting motions will be **filtered** and modified. (The special modeling requirements of computer animation that is based on motion capture techniques—including continuous skin-like surfaces and clothing—are described in Chapter 5.)

A few high-end motion capture systems can be purchased with all the components integrated, functioning, and ready to be used. This includes, for example, the computer having enough external ports to accept data input from motion sensors at adequate sampling rates. But when assembling low-cost motion control systems from parts that are purchased separately, it is often necessary to keep several issues in mind in order to have a functional system. These issues are related to the placement of the motion sensors on the actors, the stage used to capture the motion, and the type of motion capture technology.

The **number of motion sensors** used in a full-body motion capture gear—a body suit—varies from 70 in a high-performance custom system to a dozen sensors in lower-cost units. The exact **placement of the sensors** depends on many factors, such as the number of sensors available, the type of motion sensor technology being used, the type of motion that is being captured, the type of data—rotation angles or XYZ position—being sent to the computer animation program, and the type of motion constraints implemented in the computer animation software. Regardless of their number, motion sensors are attached to the body of the actor with adhesive or elastic materials, or a combination or both. Motion capture for facial animation is covered later in this chapter.

Figure 12.2.1 illustrates a minimum configuration with twenty sensors for motion capture. Seven sensors are placed in the upper body: three on the head (forehead, chin, and back of the neck base), one on the sides of each shoulder, and one on each side of the back. The four sensors in the arms are placed one on the back of each forearm close to the elbow, and one on the back of each hand. The three mid-body sensors are one on the sides of each hip, and one in the lower back or pelvis. The pelvis sensor is used to determine the position and direction of the body with respect to the floor. Finally, the six leg and feet sensors are placed one on each knee, one in the front of each shin, and one on the top of each foot. In this configuration, several important joints are not covered so their motion has to be derived with inverse kinematics techniques. Some of the subtle motion of the torso and the neck is also lost in this setup due to the

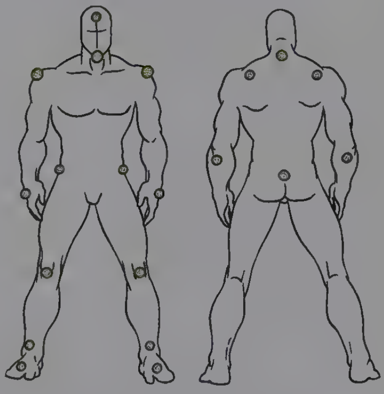
12.1.10 Davy Jones was brought to life in *Pirates of the Caribbean* with a combination of keyframe animation techniques, motion capture and motion dynamics.

limited number of sensors placed on the body of the actor.

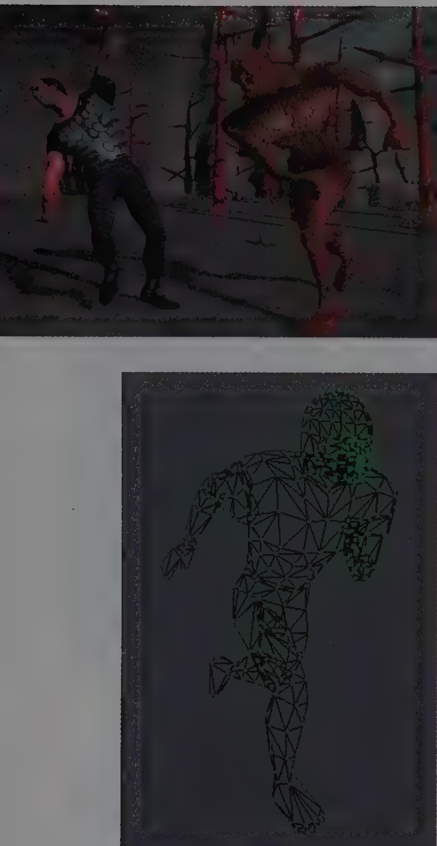
Figure 12.2.4 illustrates the implementation of a motion capture system with additional sensors, similar to the one used by actress Angelina Jolie in the mocap animated feature movie *Beowulf* (2007). Compared to the previous configuration illustrated in Figure 12.2.1, this one is able to capture a larger number of joint motions and also a greater degree of detail in the motion of the torso, head and face. A motion capture configuration like this one assumes that some of the secondary motion will be added later on top of the initial motion capture. This configuration of motion sensors includes approximately 120 sensors on the face, a dozen sensors on the head, twenty sensors on both arms and shoulders, over a dozen on chest and back, under ten sensors on the waist, almost forty sensors for wrists and hands, and under twenty sensors for legs and feet.

Real-Time Motion Capture Technologies

Several technologies are used for capturing motion in real time. These technologies have advantages and disadvantages that make each one more suitable for specific applications of motion capture—for example, capturing motion for sport games (lots of action) or for a dramatic feature movie (lots of close-up shots). Some of the factors that distinguish each technology from one another include their data accuracy, sampling rate, the freedom of motion they allow the live actors, the number of sampling points, and the number of actors whose



12.2.1 Motion capture system with twenty motion sensors.



12.2.2 Characters from *Duel* animated with motion capture (top). Below is the polygonal mesh driven by the motion capture data in *Quarterback Club Team™*. (Courtesy of Acclaim Entertainment, Inc. All rights reserved.)

motions can be captured simultaneously. The number of sampling points were covered in the previous section. Useful sampling rates start at 30 samples per second or higher. When simple motions are being captured a small capture area can be adequate, but larger areas are preferred for capturing two or more actors interacting with each other. That way, the motions do not have to be interrupted and the editing of motion is minimized. The basic technologies for motion capture include: prosthetic, acoustic, magnetic, and optical.

Prosthetic motion capture technologies provide accurate angular rotation data and are based on **potentiometers**, which are devices capable of measuring electromotive force based on the amount of energy that passes through the device as a result of the motion of a joint. However, a limitation of prosthetic motion capture is that potentiometers are usually bulky and restrict the type of motion that can be performed by the individuals wearing them. Prosthetic motion capture technologies have been around for a long time and are still widely used in medical applications that measure or simulate the motion of patients with limited motion range.

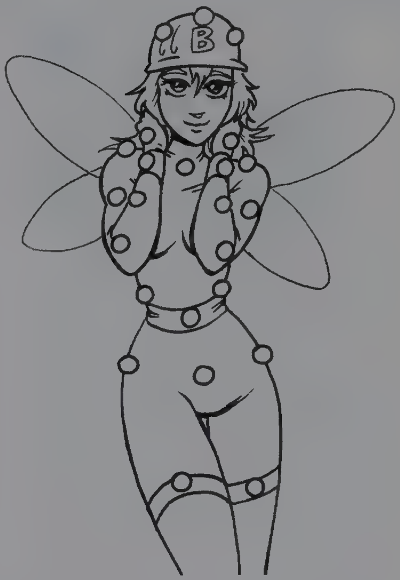
Acoustic motion capture technologies are based on **transponders** that determine their position in space by sending radio signals from each sample point. **Magnetic motion capture** techniques are based on **receivers** that detect magnetic fields. Both acoustic and magnetic motion capture technologies require stages without frequencies or noise that may interfere with the data capture in any significant way. This may include, in the former case, hard polished surfaces around the stage that may generate an inordinate amount of echo. In the latter case, this may include metallic structures in the vicinity of the stage—including metal studs inside of walls and ceilings—that may create or bend magnetic fields. In the case of some motion capture systems based on magnetic technology, it is also necessary to construct a harness above the stage to hold the wires connecting the motion sensors to the computer system and to keep them out of the way of the motion of the actors.

Optical motion capture technologies use lights, cameras, and reflective dots to determine three-dimensional positions. Optical capture of motion is convenient because the actors are virtually free to perform any motion that they are capable of. Optical technologies have also excelled at the simultaneous motion capture of more than a single actor. Figure 12.2.5 illustrates a high performance optical system that employs between 50 and 70 sensors and several cameras. It is capable of capturing the motion of two actors simultaneously. An obvious problem with optical technologies for motion capture is the fact that some of the sampling points may be hidden intermittently by the motions of the actors, especially when two or more actors are being sampled at the same time. A common solution to this occlusion problem is to increase the number of cameras used to look at the sample points. This solution can provide detailed motion, but it also increases the complexity of the motion capture process.

12.2.3 The blending of performance capture and keyframe animation produces unique stylized animation. (*Monster House*. © 2006 Columbia Pictures Industries, Inc. and GH One LLC. All Rights Reserved. Courtesy of Columbia Pictures.)

Rotoscoping

Originally developed in the early days of cel animation to align hand-drawn cartoon characters to live action background plates, **rotoscoping** is a form of time-delayed motion capture. This technique is actively used today in visual effects and character animation, to capture both rough or detailed action. Rotoscoping has two- and three-dimensional variations. **2D roto**, as it is also known, uses still frames of a live action sequence to trace manually or automatically a drawing over them. The traced drawings may be cleaned up and used as final art or as just a guide to facilitate, for example, the placement of two-dimensional hand-drawn animated characters within three-dimensional backgrounds and sets. **3D rotoscoping** uses the live action still frames to roughly align the joints of a three-dimensional character over the joints of the live action actor. The results of 3D roto can be surprisingly convincing when done well (Fig. 13.3.1).

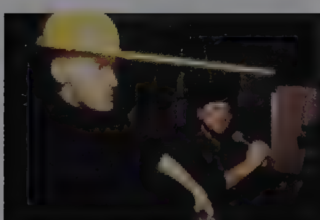


12.2.4 Approximate configuration of motion sensors used to capture the performance of Grendel's Mother in the *Beowulf* feature film.

Live Motion Control

The motion of live actors can be both captured and applied to the animated characters in real time when used for live entertainment, such as a computer-generated TV host that interacts with human actors in real time. In these cases the mocap data is used to animate the characters which are instantly matted into a live video feed. In many of these instances, the animated characters are cartoons with geometries simpler than those used for feature productions (Fig. 1.4.5). The goal of the motion capture process in this type of application is capturing the theatrical broad-gesture motion that can bring a cartoon character alive; it is less interested in details and nuances (Figs. 12.2.6 and 12.2.8). Generally speaking, live cartoon characters look best when animated with motion capture that is a bit exaggerated and spirited, the kind that can be delivered by experi-

12.2.5 Two actors, wearing optical sensors for motion capture, engaged in a simultaneous two-person capture. (Courtesy of Acclaim Entertainment, Inc., Advanced Technologies Group.)



12.2.6 Two performers do a live animation of *Moxy*, the animated character shown in Figure 1.4.5. One of them is in charge of creating the voice of the character, and the other creates the motion of the head and hands with a simple motion capture system. (Produced by Colossal Pictures in association with the Cartoon Network. Courtesy of Colossal Pictures.)

enced puppeteers and actors. These professional live performers are able to transmit emotion and expression through motions captured by a cold input peripheral. Certain situations call for more than one individual manipulating the input peripherals for controlling the motion of the character. These peripherals may include, for example, one for the lips, one to set the motion path that controls walking, and one for limb joint rotations.

Editing the Captured Motion

A motion capture session usually results in several tracks of motion that control different parts of the animated character. Each track is assigned to a channel that usually controls a specific joint or a group of vertices. The motion data in the channels is generally displayed by computer animation software as function curves (Fig. 12.2.9). Once all the captured motion data has been cleaned and ported into the respective channels then it can be attached, in most mocap done today, to different joints in the inverse kinematics skeleton of the character.

There are different strategies for how the mocap data is previewed on the set. Most systems today offer a limited preview of how the mocap data is being sampled and how it maps onto the different channels of the animated character. This results in an efficient front-end process that nevertheless requires a fair amount of work to be done in the back-end, a constant in the production of computer animation. Most high-end mocap-based features to date have been done with limited or minimal preview. A few software developers are trying to satisfy the creators' desire for an instant and complete preview of the mocap session in the ideal form of the director being able to view the virtual actors in real time through a hand-held camera. We are still not quite there, but judging by how fast the technology is evolving, it is possible that mocap systems with full real-time preview capabilities will be available in the near future.

Some planning and preproduction work is required to make sure that the motion capture data can be used on the IK skeletons. These have to be structured so that the subtle motion captured in the form of multiple XYZ rotations has a clear place to go and will not be applied to the wrong joint. In addition, some **motion blending** between channels may be necessary to improve the look and feel of the motion. Blending motion between channels preserves the relevant mocap information and facilitates layering of additional animation such as lip-syncing, hand motions, and rule-based facial expressions.

Channel Animation

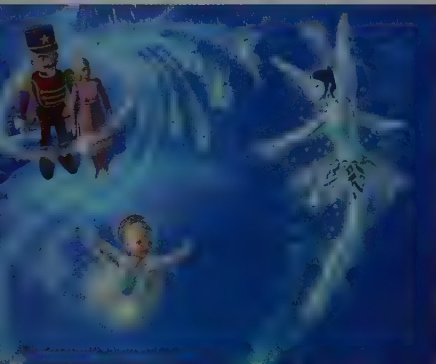
Channel animation allows the collection or capture of all kinds of information in real time through a variety of input peripherals that are attached to a computer. The data—motion capture or other—contained in the channels can be used to control non-character

aspects of the animation, such as the intensity of a light source, the density of a texture, the force of gravity, or the speed of an object. The sets of data brought into the system are assigned to one or several **channels** in the **animation score** and used to drive the aspects of the animation controlled by those channels. For ease of work, the captured data is displayed in the form of parameter curves, also called **function curves** (Fig. 12.2.9).

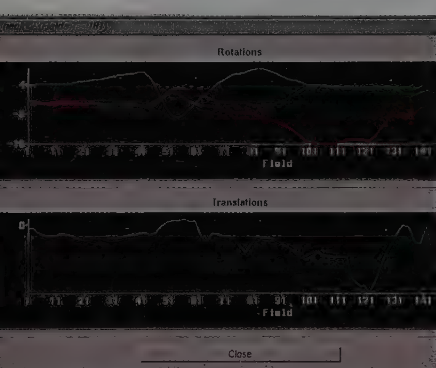
All kinds of input peripherals can be used to input time-based information to the animation software. This includes **peripheral input devices** such as a joystick, a microphone, a music keyboard, a trackball, a variety of motion capture gear, even a track wheel systems for controlling the virtual camera (Fig. 12.2.3). In all cases, a **device driver** is necessary for the animation software to be able to communicate with the peripherals.

The basic process of channel animation starts by identifying the active input devices and assigning each of them to one or several channels based on the number of degrees of freedom, rotations, and translations that the device is capable of. A joystick with one button, for example, has three degrees of freedom because it can move along two axes and the button can be clicked. Motion capture gear can generate dozens of channels depending on the number of position points—each with XYZ degrees of freedom—the gear is built with,

12.2.7 The Gollum character was animated by WETA Digital, with a combination of keyframe animation and motion capture techniques based on the performance of actor Andy Serkys. The character's emotions and facial expressions were animated using the Facial Action Coding System (Figure 10.3.6). (*The Lord of the Rings: The Two Towers* © MMII, New Line Productions, Inc.™ The Saul Zaentz Company d/b/a Tolkien Enterprises under license to New Line Productions, Inc. All rights reserved. Photo appears courtesy of New Line Productions, Inc.)



12.2.8 The motion capture data generated by dancers from the American Ballet Theatre was applied to some of the characters in *Barbie and the Nutcracker*. (BARBIE and associated trademarks and trade dress are owned by, and used with permission from, Mattel, Inc. © 2003 Mattel, Inc. All rights reserved.)



12.2.9 Detail of a motion editor program window showing the rotation and translation data channels for one bone in an articulated skeleton. (Courtesy of Acclaim Entertainment, Inc., Advanced Technologies Group.)

which can range from a dozen to hundreds. Once each degree of freedom in the device is assigned to a channel, the second stage of the channel animation process consists of assigning each channel to the motion of an object. Figure 12.2.11 shows dialog boxes to control the process of blending channels and **retargeting motion**.

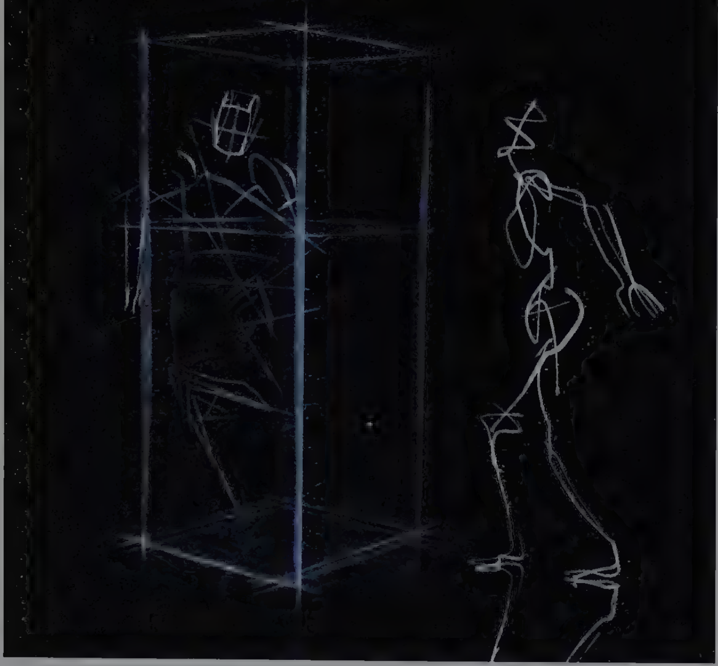
MIDI Output

Using music as the source for motion is another effective method for creating channel-based and procedurally-driven animation. Unlike the digital recording of a specific sound or music, the **Musical Instrument Digital Interface** protocol, generally known as **MIDI**, contains channels with information about how to play or interpret specific musical notes. MIDI contains information about the actual musical notes and keys, the length, the pressure, the instrument, and other parameters associated with musical interpretation. MIDI is commonly used to feed music synthetizers that execute the MIDI instructions in real time to create sound. Figure 12.4.6 shows an example of MIDI data used to procedurally drive the animation. In this case the MIDI data is processed with a proprietary software for generating sequences of motion for the given music. This software uses standard MIDI parameters like note, volume, pitch-bend, modulation, and sustain-pedal, to output position, rotation, scale, and light intensity information. The most interesting motions are obtained by using combinations of several algorithms to map the MIDI date to three-dimensional data. In the case of MIDImotion software, for example, the process is straightforward. Low notes can be mapped to large, slower-moving objects, while higher notes can be mapped to smaller, faster-moving objects. Drumstick motion can be derived by using a technique reminiscent of robotic motion planning. Additional motion such as ball trajectory and object impact can be calculated using basic physics. If the music is changed, the motion is automatically regenerated. Finally the generated parameter channels are used by commercial animation software via plug-in technology.

12.3 Dynamics Simulations

Dynamics simulation techniques, also called **motion dynamics**, generate realistic motion of rigid body objects or fluids by simulating their physical properties and the natural laws of physical motion. Motion dynamics techniques take into account the characteristics of solids (for example, weight, mass, inertia, and flexibility), liquids and gases (for example, density, cohesion, viscosity, even stickiness), as well as external forces such as temperature, speed, pressure, friction or gravity, and even collisions with other objects. Dynamics simulations can be combined with other advanced animation techniques such as inverse kinematics and simple keyframe animation.

A **rigid body dynamics** simulation calculates the motion of



12.2.10 Still frame from a multimedia art installation in which motion-captured human dancers represented as hand-drawn figures move in three-dimensional space. Short motions, or phrases, were performed, captured, and later augmented and placed within the virtual choreography as a whole. The positions of light-sensitive sensors attached to key points on the performer's body are recorded with optical cameras that record these points as coordinates in a three-dimensional data set. The motion capture data was manipulated with Character Studio, using its patented footstep-driven keyframe approach. (*Ghostcatching* © 1999 Bill T. Jones, Paul Kaiser, and Shelley Eshkar.)

objects through time by providing the software with some of the physical properties of an object—mainly its mass—as well as some information about the forces applied to the object (Fig. 12.3.1). The **mass** of an object is established by the product of the **volume** of the object and its **density** (mass = density × volume). Forces have a specific **strength** or **intensity**, and a **direction**. In simple terms, a dynamic simulation calculates the **acceleration** experienced by an object with a certain mass when a **force** is applied to it (force = mass × acceleration). The motion of objects is calculated by using the effects of acceleration on the object over distance and time to define the **velocity** and positions of the object through time.

A **fluid dynamics** simulation can simulate the motion of non-solid materials such as liquids and gases through time and space, using different pressures and temperatures to visualize changes in density, mass, and **viscosity**. Some of the striking stresses and flow effects revealed by fluid dynamics simulations include **turbulence** with its trademark **vortexes** and **swirling**, perturbation, recirculation, compression, expansion, and diffusion. Fluid dynamics simulations are commonly calculated on **particle streams** or blobby surfaces. Recent areas of technical development in the area of fluid dynamics simulation include non-Newtonian fluids, such as emulsions and sticky liquids, and the effect of fluids on absorbent solids such as sponges.

Dynamic simulations are calculated based on a particular length of real time and then sampled at a specific rate of frames per second. Ideally the dynamic simulations should be sampled at a minimum of 24–30 frames per second, which is the rate at which film and video are recorded and platform games played at. Dynamic simulations are run by default on all of the elements present in the three-dimensional

(*Speedy* characters Shakes and Speedy. © Tim Mostert, 2008.)

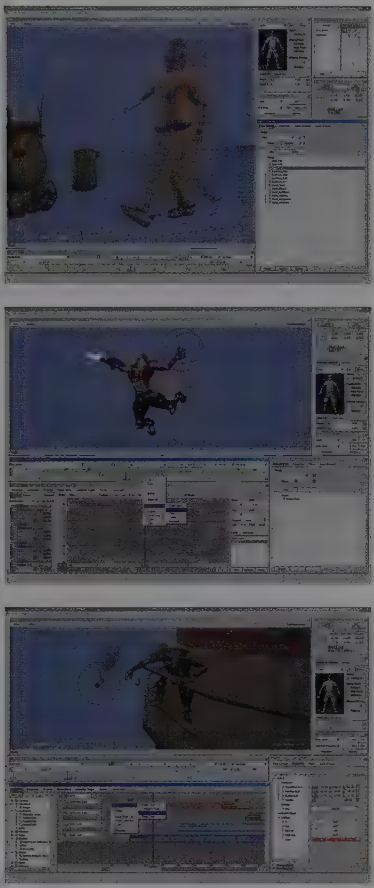
environment. Cameras and lights have to be turned off in the simulation so that they are not affected by it. Otherwise, the cameras can be moved by the simulated forces, and the moving lights can influence the final simulation.

Physical Properties of Objects

Mass is the physical property of an object that most influences a dynamic simulation. As mentioned earlier, the mass of an object can be easily determined based on its volume and density. The volume of three-dimensional objects can be automatically calculated by most computer animation programs, so the density of an object is often the only value that animators are required to provide in order for the software to calculate the mass of the object. Other characteristics of the object can also contribute to the realism of its motion. **Elasticity** and **stiffness**, for example, can be used to define the rigidity or flexibility of an object especially at the times of collisions (Fig. 12.3.3). **Rigid objects** do not bounce far from a collision, and their surfaces do not move much—if at all—after the collision. A solid ball made out of steel is an example of a rigid object that is extremely stiff and, as a result, does not deform when it hits most surfaces. But the steel ball bounces off the surface because it is somewhat elastic. **Flexible objects**, on the other hand, may bounce far away from the collision point. The surfaces of flexible objects also deform significantly as a result of the collision and may keep moving moments after the collision took place. Objects made of hard rubber and gelatin, for example, well illustrate the range of flexibility in characteristics of elasticity or stiffness. A solid ball made of **hard rubber**, for example, is a flexible object that is elastic. As a result it bounces hard when it hits a surface, but it does not deform much because it is quite stiff. A solid sphere made out of **gelatin**, on the other hand, is not elastic at all. As a result it bounces little—or not at all—when it hits a surface, and it deforms greatly because it is not stiff at all.

The ability of flexible objects to absorb the impact of a collision by deforming their shape is usually controlled in dynamic simulations by applying the forces to a **flexible lattice** that controls the vertices in the object. With this technique, the bending and deformation of the surface of an object is filtered depending on how the lattice points control the vertices of the object. Some computer animation systems convey stiffness with functions that simulate the effect of having **springs** between the vertices on the surface of the object. Springs have a natural rest position that they always return to after being stretched. Springs will move back and forth between the **stretched position** and the **rest position** until the initial balance is restored.

In some cases, the stiffness of rigid objects and the force of a collision are such that the real object is incapable of absorbing the force of impact and breaks or shatters. A dynamic simulation of an object breaking is much more complex than the simulation of objects that



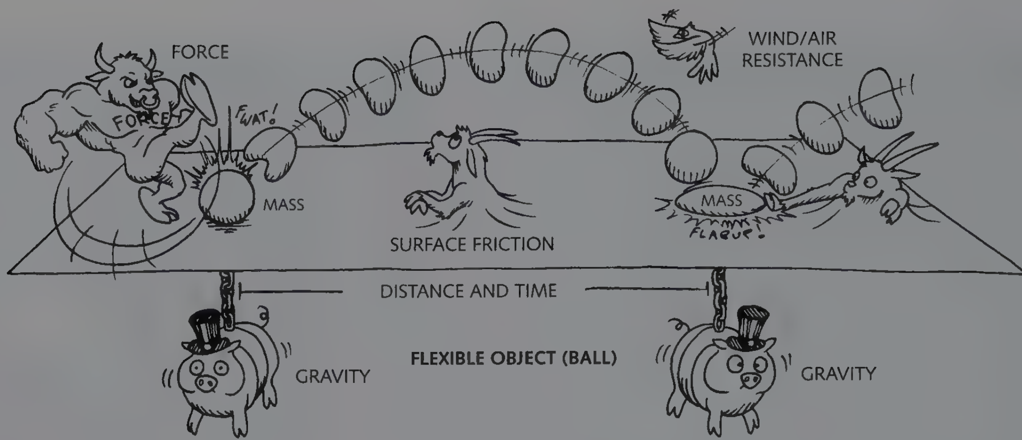
12.2.11 Motion-editing software can be used to create a catalog of character poses by copying-and-pasting matching poses from any track in the control rig. Poses are character-independent and can be retargeted to any character (top). Some inverse kinematic controls allow to retarget motion from any source to any character. With control rigs one can animate over a motion source without altering the original animation. This provides animators with great control over the animation with minimal editing (middle). Two overlapping constraints that control the same character, for example, can be blended into new motion (bottom). (Motion Builder dialog box courtesy of Kaydara Inc.)

do not break because, in essence, the results of the collision would have to be applied to thousands of fragments instead of to just one object (Fig. 12.3.8). In addition, issues of structural composition, brittleness, randomness, and chaos theory would have to be calculated to make the dynamics simulation as realistic as possible. In most productions—except those of a scientific nature—it would make more sense to fake the shattering of an object instead of simulating the motion dynamics of the event. One way to approximate the shattering of an object as a result of a collision consists of applying several forces to the object and approximating its values by trial and error until the resulting motion looks like it was the result of a dynamic simulation. This method, however, requires at least two models of the same object. One of the models is whole and is used until the collision takes place. The second model is shattered prior to the collision but all its parts are kept together, and it is used only after the collision. For example, an object can be thrown into a collision course with a specific linear force. But when the object reaches the collision point then the initial force is cancelled, the initial model is replaced with the shattered model, and one or several new point forces are applied to the second model in order to throw it away from the surface.

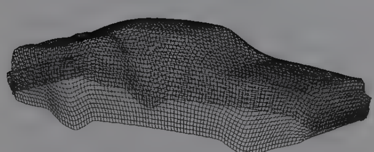
12.2.12 The personality of a character comes through in his performance facing different situations and challenges. (*Gopher Broke* created by Blur Studio, Inc.)

Types of Forces

Many types of forces can be simulated with motion dynamics techniques. The basic forces include linear forces, point forces, and coni-



12.3.1 The dynamics simulation of flexible (ball, above) and rigid objects (anvil, opposite page) takes into account the mass of the objects and the forces that propel them, as well as the forces of gravity and friction, and the distance and time traveled by the object.

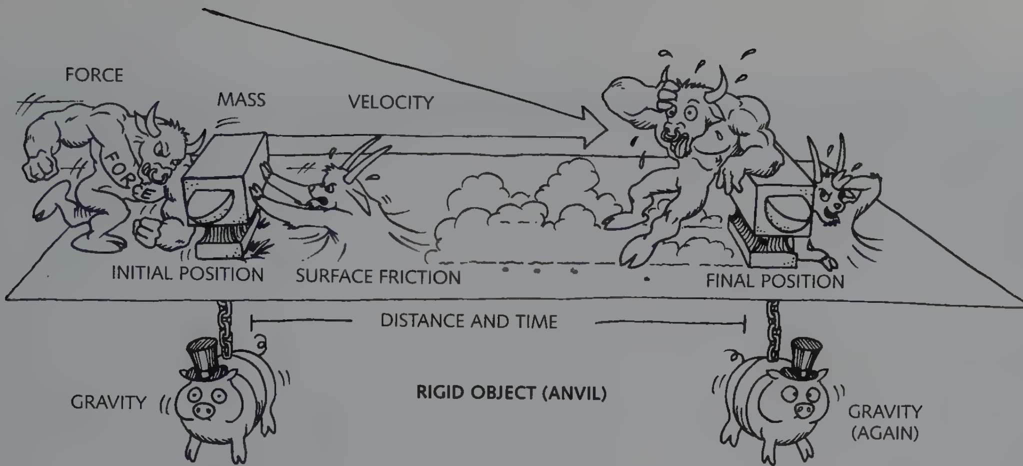


12.3.2a The resolution of the dynamic mesh used to model a Lexus car flexible surface was a total of 6,000 points arranged in a grid of 60 columns by 100 rows. This high resolution was necessary to represent the detailed image map. (Animated by Mark Henne. Courtesy of Rhythm & Hues Studios.)

cal forces. These basic forces can be used in combination with one another to create more complex forces. A **linear force** is unidirectional; it has one intensity value, and is traditionally represented with a vector. Linear forces include those of wind and gravity, punching, or throwing. A **point force** or **radial force** travels like rays in all directions and is best illustrated with the forces released by a bomb that explodes in all directions. A **conical force** resembles a collection of linear forces that spreads out of a single point resembling the shape of a cone. When these forces impact a surface, they are strongest at the center of the impact area and weaker at the edges. The forces created by a fan, for example, are conical forces.

Simple forces can be combined with each other to produce variations of complex forces. Figure 1.2.8 illustrates the complex turbulent forces that are behind severe storms. In this case, the evolution of small clouds into giant storms was simulated with fluid dynamics. The simulation of a storm development can be a useful tool for understanding storms' behavior, and even to predict when a severe one may appear based on the conditions that usually lead to the development of similar storms. The equations used to simulate storms or other time-dependent events can be solved by first specifying the initial values of wind velocity and direction, temperature, pressure, and moisture at selected locations within a specified, three-dimensional, rectangular region of the atmosphere. This region is often called the **simulation domain**. The changes in these values are then computed every few seconds over a time span of several hours. Due to the extremely large amount of data that is necessary to achieve a realistic simulation of this kind, it is common to perform the computations on a powerful supercomputer or a parallel processor. Performing the same type and number of calculations on one of today's microcomputers—or even a low-end super-microcomputer—could take several weeks and would be impractical.

Forces can be applied locally or globally. **Local forces** affect only one object or one joint, while **global forces** affect all the



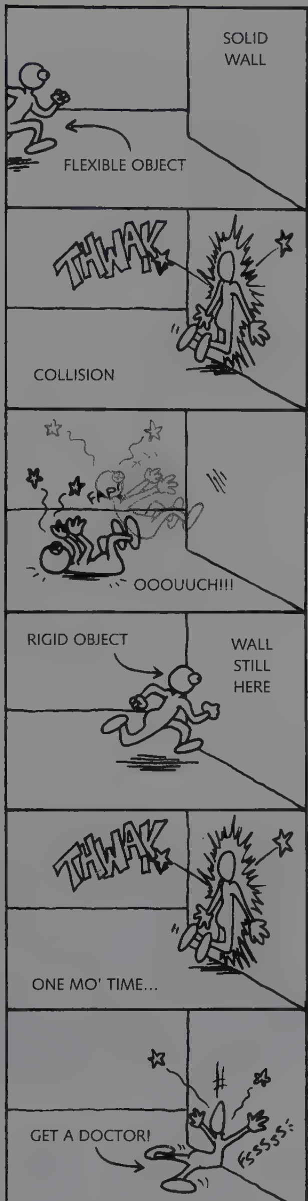
objects in the three-dimensional environment. The force of the Earth's **gravity** is a good example of a global linear force. The force of one ball pushing just another ball on a billiard table is an example of a local force. Forces can also impact, attract, or resist objects. **Impacting forces** push objects away from the source of the force, like wind does. **Attracting forces** pull the objects in like magnets. **Resisting forces** offer resistance or opposition to objects moving through the three-dimensional environment. Examples of resisting forces that can slow motion down include friction and viscosity. **Friction** happens when one surface rubs against another. All spaces, unless they are a vacuum, have some amount of **viscosity** or **environmental density** that facilitates or impedes the motion of objects. In underwater scenes, for example, moving objects encounter more resistance from the density of water than they do from the density of air (Figs. 12.3.4–12.3.6).

The image shown in Figure 12.3.2b is a good example of a pioneering motion dynamics simulation created for a TV commercial. The animation of the car was created with a combination of techniques including the deformation of a flexible lattice that controls the vertices in the surface by applying wind forces to it. Each of the points on the lattice had a specific value of mass assigned to it, and they were all connected with simulated springs (Fig. 12.3.2a).

The wind forces applied to the lattice were timed so that they would start at different points in time, and were focused on different areas of the lattice so that the motion would look as if driven by natural wind forces. The forces were applied in a variety of ways including linear, conical, and turbulent, all of which had intensity and directional parameters that were animated throughout the sequence. The forces had variable intensities so that they were strongest at the center and weaker at the edges.

The project pictured in Figure 12.3.2 is also a good example of a motion dynamics simulation that included **shortcuts in the simulation** to fit both the production deadlines and a limited budget.

12.3.2b A new Lexus car model is revealed after the cloth-like surface is animated so that it ripples and flies away. The image of an old car model was texture-mapped on the rippling surface. (Animated by Mark Henne. Courtesy of Rhythm & Hues Studios.)



12.3.3 During a collision flexible objects are elastic, but rigid objects do not bend much.

Early in the production process of this pioneering animation it was determined that it would not be possible to use collision detection techniques to keep the flying surface from colliding with the body of the car or from penetrating its space. This was due to techniques required, production deadlines, and budget issues. Instead, the simulated layer was kept away from the car by applying to it radial forces from below as if it were blowing in an upward direction. But the shortcuts used in this simulation were balanced by performing the computation of the wind forces at intervals shorter than one frame. This was necessary due to the fact that in one cycle of this dynamics simulation the spring forces attached to each vertex on the mesh only affected the immediate neighboring vertices. For the forces to spread to several vertices and—as a result—for the cloth to ripple with detail, it was necessary to simulate several cycles between frames so that the rendered images contained rippling that propagated throughout several vertices in the mesh. See Figures 1.2.9 and 5.5.12 for additional examples of cloth simulation with collision detection.

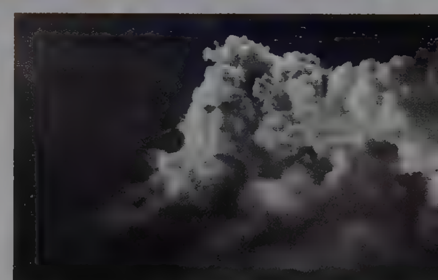
Collision and Collision Detection

The motion that results from a collision can be calculated in a variety of ways. The simplest approach consists of aiming the collision forces at the center of the object, and assuming that the mass is distributed evenly throughout the object. Other approaches can be used to simulate richer and more realistic motion, but they are also much more time-consuming to calculate. One of these approaches starts by determining the accurate position of the **center of mass** of the object, as opposed to using the geometric center of the object as the center of mass. The **distribution of mass** is also calculated. Symmetric objects usually have a balanced distribution of mass, but irregular objects with an uneven distribution of mass such as meteorites tend to have unpredictable motion. When forces are applied to objects on parts other than the center of gravity they tend to produce motion that is not linear. These forces are call **torques** because the motion they produce is in the form of rotations or torsions with varying amounts of **rotational velocity** and acceleration, and changing orientations (Fig. 12.3.8).

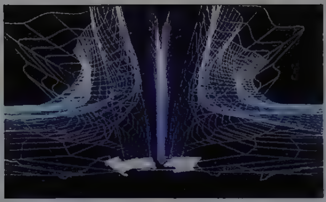
One of the most interesting and useful applications of motion dynamics animation techniques is detecting collisions between the objects being animated. Real objects react naturally to a collision by deforming and changing the direction and speed of their motion, and even breaking. Simulated three-dimensional models, however, will naturally ignore other objects that penetrate their space unless collision detection techniques are used. Using **collision detection** techniques on complex geometry can add a lot of processing time and expense to a scene because they must constantly check the position and dynamic properties of objects in order to avoid having objects overlap. A simple, inexpensive alternative to collision detec-

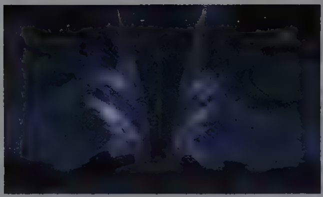
tion for small animation projects that do not involve motion dynamics techniques is previewing the animation—in the form of a motion test, for example—and detecting the collisions visually. The correct positions of the overlapping objects can then be approximated manually, and the sequence can be previewed again with simple keyframe animation techniques. One of the problems posed by visual detection of object collision is that it may require a lot of time in scenes with a multitude of objects.

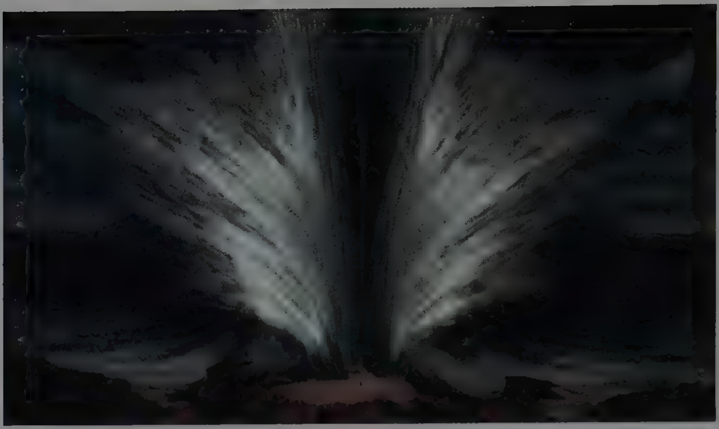
Automatic collision detection is convenient because it frees the animator to do other tasks that are more important than detecting collisions visually. Automatic collision detection is also usually faster and more accurate than visual collision detection. There are many techniques for automatic collision detection, and a number of them are provided by turnkey commercial software. A common method for doing a first pass collision detection test consists of using rectangular or spherical **bounding volumes** or surfaces (Fig. 4.6.4). This method can save thousands of calculations by simply determining whether the boxes intersect at any point. If the bounding boxes intersect then a second stage collision detection test can be performed with the objects themselves to determine whether they intersect. A third stage collision detection test usually consists of checking the polygons of one of the intersecting objects against the polygons, or even edges, of the other intersecting object. In cases when the collision detection test is positive, then a response to the collision can be animated with motion dynamics techniques.



12.3.4 A fluid dynamics simulation of clouds that displays great rendering craft. Shaders were fine-tuned to render realistically the slow turbulence and changing nature of clouds. (Images courtesy of The Mill.)







12.3.5 The walls of water were created with dynamics simulations and particle systems. (Photographs from the motion picture *Prince of Egypt*™ © 1998 DreamWorks L.L.C., reprinted with permission by DreamWorks Animation.)

12.3.6 (Opposite page) A low-resolution BMW model was used as a collision object with the fluid dynamics simulation, which was divided into a few parts with a maximum of 1.5 million particles each. Footage of the Skogafoss waterfall in south Iceland was used as reference to develop the water shaders, set up the lighting configuration, and analyze the motion of splashes, sprays, and foam. The water meshes calculated with the RealFlow software were imported into SoftimageIXSI, and rendered along with High Dynamic Range images and a high-resolution BMW model for shading the refractions and reflections in the water. All layers plus some of the reference footage was combined in compositing. (Postproduction by Pictorion das Werk, Germany. Fluid simulation by Christian Laskawi, shading by Sebastian Weidner, and compositing by Ben Turner. Images courtesy of Next Limit.)

The approach for collision detection that relies on brute computing force consists of testing all the objects in the environment against each other. In an average computer animation production, this approach only makes sense when the motion in the scene is such that most of the objects are expected to bounce into each other. But a simpler and more economical method for collision detection can be implemented by identifying the **obstacles** that the moving object is likely to encounter along the **collision path**. Figures 6.6.2, 12.3.7 and 12.3.8 show colliding objects that break into smaller pieces, the latter image based on a finite element analysis simulation.

12.4 Procedural Animation

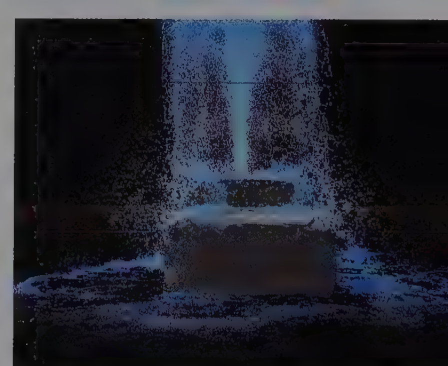
Procedural or **rule-based motion** techniques animate the elements in the scene based on a set of procedures and rules that control motion. Rule-based animation has a wide range of applications that includes the animation of natural phenomena, flying birds, growing plants, fantastic life forms, and humans dancing or gesturing. (See Chapter 5 for more information on modeling plants with procedural techniques.)

Particle Systems

One of the most popular forms of procedural animation is exemplified by the animation of particle systems. Animation with **particle systems** recreates the motion of particles that follow some generally defined motion. In the majority of computer animation programs, the particles themselves do not have a specific shape, but they can be used to control other objects or attributes. When particles are used to recreate the light of fireworks, for example, they represent a point of light with a variety of attributes such as intensity, flickering, and tail-tracking values (Fig. 5.5.6 and 12.3.4, 12.4.1, and 13.6.1–13.6.3).

Particle systems are used to represent dynamic objects that have irregular and complex shapes, each with its own behavior. Particles

have a life span during which they are created, behave a certain way, age, and die. Particles can also be used to control the motion of three-dimensional models—such as snow, water, or even a flock of birds—and to animate the growth process of plants by encoding their characteristics in a series of rules that can be used as the basis for a simulation (these methods are described in Chapter 5).

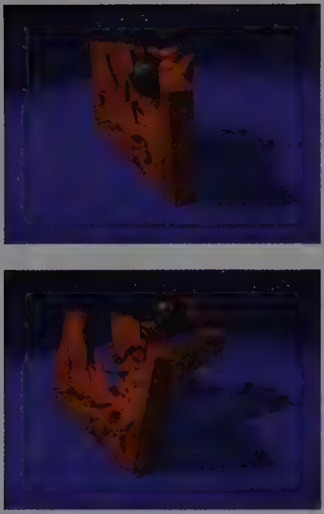


Flock Animation

There are many different strategies to generate flock animation. In most cases, the behavior of the birds in the flock is contained in a series of rules that constitute the computer model that simulates the flock animation. These rules control all the variables involved in the behavior of the flock. These rules include, for example, whether the flock has multiple leaders or a single leader and, if multiple, in which pattern does the rest of the flock follow the leader. Some of the basic variables provided by most computer software to control the motion of flock animation include the way in which the members of the flock move towards a target, how they avoid obstacles, and how they relate to other members in the flock as the flight conditions change throughout time.

Animating a flock of birds with rule-based techniques is a more practical alternative than using keyframe animation. Flocks can be sim-

12.3.7 Dynamics simulation used in *The Moment* to animate a crash of two cars. (© 2007 Filmakademie Baden-Württemberg, Verena Fels, Csaba Letay, Heiko Schneck, Hendrk Panz.)



12.3.8 The animation of this adobe wall struck by a wrecking ball is simulated with linear elastic fracture mechanics. The simulation determines where the cracks should start and in which direction they should propagate after analyzing the stress tensors that are computed over a finite element model. As the animation develops, the software remeshes the geometry to create the dynamic fractures. (Images courtesy of James F. O'Brien, W. Wooten, and J. Hodgins. © 1999, Georgia Institute of Technology.)

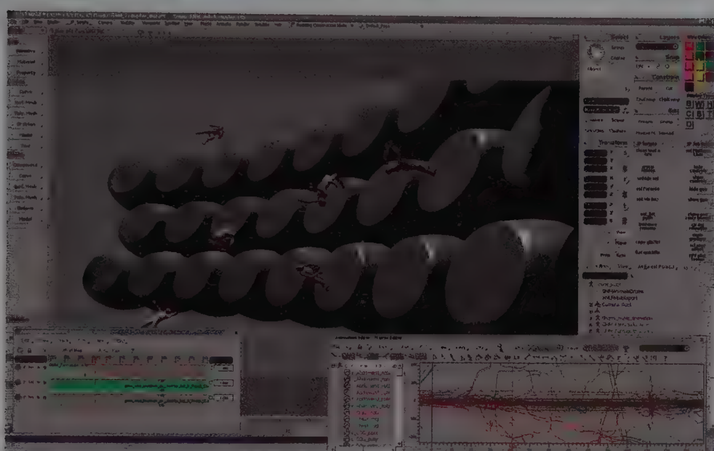
ulated with particle systems so that each particle in the system represents a bird. Each bird in the flock moves according to the laws behind the physical simulation, its own perception of the environment formalized by the rules of the system, and by a series of parameters defined by the animator. The overall motion of a flock can be represented as the result of the behavior of each individual bird and the interactions between them. A common strategy to recreate the flying behavior of each bird is based on rules that simulate some of its perception and the action of flying. Once the model is expressed in terms of rules then several birds can be simulated and allowed to interact with each other.A significant difference between particle systems and flock animation based on particle systems is that in flock animation the particles are replaced by three-dimensional models; they have orientation, and they also have more complex types of behavior.

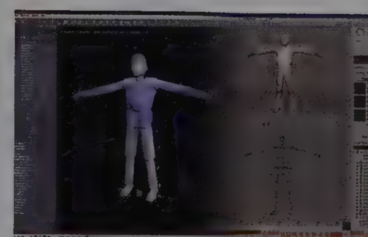
The behavior of flocks is determined by the internal conditions of each bird in the flock, and also by the external conditions that affect the flight of each individual bird and the flock as a whole. Birds in the flock present many forms of behavior and goals. Common behaviors and goals of flocks include, for example, avoiding collisions with other birds in the flock or objects in the environment, matching the speed of nearby birds, and staying together. Each of these behaviors requires a specific acceleration and direction.

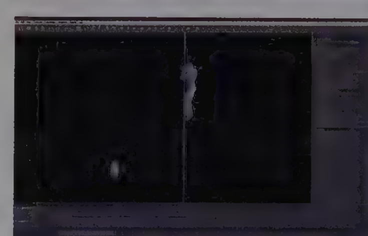
When the goals of dozens of birds are in play it is necessary to arbitrate all the individual requests. The flocking model used to create Figure 1.4.1 employs a variety of techniques to arbitrate independent behaviors. These techniques are based on a prioritization of all the component behaviors and their acceleration requests. Depending on the situation, different requests get a higher priority. For example, maintaining all the birds in the flock together can receive a low priority if the flock is about to collide with a large obstacle.

Computer animations of flocks can also be created with a combination of particle animation and keyframe animation. This is especially

12.3.9 Complex human behaviors built in real time with the Endorphin physical simulation software for the *Guinness Music Machine* commercial. A variety of forces are applied to characters to produce primary and secondary behaviors that can be exported to a keyframe animation system. (Images courtesy of The Mill.)







useful in cases when the computer animation software does not provide a full-fledged rule-based system. The flock animation in Figure 1.4.1 was controlled mostly with simple parameters that are adjusted at the keyframes, but the relation between flock members cannot be specified in the form of rules. In that case the process started by creating a number of particles that will represent the members of the flock as they move from a **source mesh** for particles to a **destination mesh**. The precise path of the motion can be controlled by the magnitude and direction of a simulated force of gravity, and also by the way in which the particles select the points (or vertices) on the source mesh to leave from and the points on the destination mesh to arrive to. The distribution of the particles on both meshes can be easily controlled by the shape of the meshes and by whether the source and destination meshes have the same shape and number of vertices. The distribution of particles can also be controlled with numerical values that randomize the mesh or by concentrating most flock members on

a small group of vertices. The behavior of the flock as it moves from one point to another can be controlled in this example by numerical values that specify the amount of back-and-forth change, or jittering, in each of the geometric transformations, and the shape of the models being controlled by the particles. The shape of the three-dimensional model that is controlled by the moving particles can be specified in the form of one or several key shapes. Other approaches to flock animation are illustrated in Figures 12.4.2, 12.6.1 and 12.6.2.

12.3.10 Final result of a fur generation procedure (top) and four steps in the process: a monochrome map to control length of the hairs, a few initial growth guides (in green), the guides and the fur, and combed guides and fur. (Images courtesy of Hans Rijpkema, Rhythm & Hues Studios.)

12.4.1 (Opposite page) Simulation of a fireball for an effects escape sequence, also showing previsualization with the proxy models and simplified rendering. (Shrek™ and © 2001 DreamWorks L.L.C.)

Goal-Oriented Animation

Some computer animation systems are capable of automatically choreographing the motion of an animated character based on a specific goal that has to be achieved. The animated characters in goal-oriented systems range from a simple robot arm to a fantastic creature or a human-like character. The goal for the characters can be as simple as turning the head towards the light, or as complex as grabbing an object with the left hand, passing it to the right hand, and running out of the room while avoiding all the obstacles along the way. **Goal-oriented** computer animation is also often called **intention-based** or **automated** animation. Goal-oriented animation has its roots in the fields of robotics and expert systems, where computer systems are designed so that they can be as autonomous as possible, including the ability to plan different strategies to achieve a goal, evaluate the results, and continue to develop the strategies that were successful while avoiding the strategies that lead to failure.

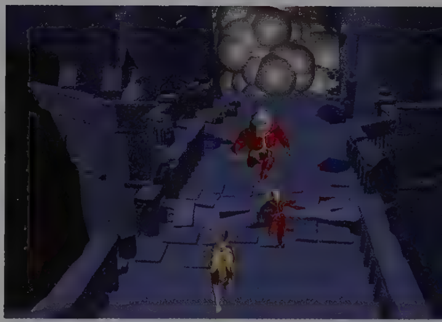
Goal-oriented animation techniques provide animators with insightful analysis into how humans move to accomplish a certain action. Even though these techniques still belong by and large in research laboratories, their usefulness is turning some of them into commercial products, especially as powerful features of game engines that seek to populate games with **autonomous characters**. The most important component of a goal-oriented animation system is the set of rules and procedures that allows a character to analyze and evaluate its environment and to determine the best way to achieve a goal, usually by reacting with motion, gestures, and manipulations of objects in the environment. Many goal-oriented animation systems also include an inverse kinematics module to deal with the position of jointed figures, and/or a motion dynamics module that deals with basic issues, including collision detection, weights, and forces.

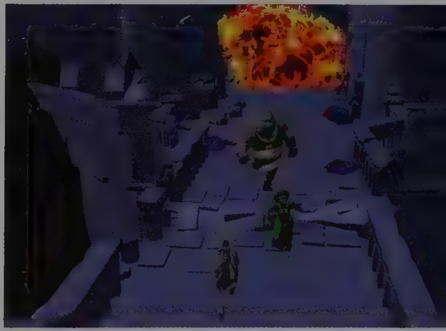
Goal-based computer animation systems include the **codified procedures** that are necessary to analyze a goal, break it into tasks, evaluate the environment, predict and try to avoid potential obstacles, recover from mistakes, develop new strategies as a result of those experiences, and ultimately achieve the goal. Most existing goal-oriented computer animation systems specialize in a specific type of goal or a specific type of motion; otherwise their tasks would be too complex to implement. (When the complexity of goal-oriented ani-

mation approaches the complexity of human motion it is often more practical to record human motion itself directly on video or film.) Some goal-oriented systems, for example, specialize in simulating human gaits and other forms of multilegged locomotion. Others can animate hands grasping and manipulating objects, or even body gestures and facial expressions.

One of the main tasks of goal-oriented animation system is determining what sequences and paths of motions are necessary to achieve a certain goal. Finding an optimal motion path that will allow the goal to be completed involves testing for collision detection, angles of motion, and grasp ability of limbs. Establishing a **sequence of motions** involves determining how many steps are necessary to complete a motion and what is the optimal order of execution. Simple goals that involve motion usually translate into simple motion paths and simple motion sequences. But as the goal increases in complexity so do the paths and sequences of motions necessary to complete the goal.

One of the biggest challenges of goal-oriented computer animation is to deal effectively with complex sequences of motions, both in terms of being able to complete the tasks and achieve the goal, and also in terms of producing natural motion when human figures are animated. For this reason, most are based on some sort of **motion planner**. The systems that animate characters that grasp and manipulate





12.4.2 A villain character from *Dragon Hunters* with special powers to disintegrate into a flock of bat-like creatures and reassemble at will. (*Dragon Hunters* © MMVII Futurikon Films, Trixter, LuxAnimation, France3 Cinéma, RTL-Tvi, in coproduction with Mac Guff Ligne.)

objects use a **manipulation planner**. In addition, goal-oriented systems include kinematics or dynamics techniques for calculating motion. The manipulation planner used to create the sequence of images in Figure 12.4.3, for example, defines paths in terms of transit and transfer paths. In this case, the goal for the animated figure was reaching for the glasses and wearing them. The task of the animator is limited to selecting the object that has to be moved and the location where the object has to be repositioned. The motion planner of this goal-based animation system determines that the character has to use both hands in order to complete the action. Not too many individuals can grab a pair of lenses and put them on with just one hand. The **transit paths** define the motions of the character without the objects being manipulated—for example, getting the arm to a position from which it can reach the object. The **transfer paths** define arm motions that also move the object. Transfer tasks are generated by analyzing and planning the motion of the object from its initial position through the completion of the goal. During the calculation of the path, the manipulation planner identifies all the possible ways of grasping the object and the configurations of the object requiring a grasp or regrasp.

The majority of motion planners have a simplified set of rules that specify how motion takes place in general, and which motions are allowed in particular. The animation system illustrated in Figure 12.4.3, for example, allows only the arms of the animated character

to touch the objects in the environment that are to be grasped. Objects in the environment that are obstacles can only be touched for the purpose of achieving static stability. In the interest of efficiency, most goal-oriented systems limit the number of possible motions and grasps, or types of static and dynamic obstacles that are considered, or the number of possible solutions to situations when collisions occur.

A few goal-oriented animation systems attempt to automate the animation of human figures based on instructions given in plain English—or other human natural languages—as opposed to special-purpose animation languages. One example of such an approach is illustrated in Figure 12.4.4. This animation system is able to automatically generate and animate conversations between human-looking figures. The conversations include motions such as facial expressions and arm gestures. The facial expressions are associated with motions of the head, eyes, and lips. The arm gestures include coordinated motions of the arms, wrists, and hands. The rules underlying this system are based on the relation between verbal and nonverbal communication. The combined meaning of speech, body language, and facial expressions can result in animated characters that are consistent, believable, and somewhat autonomous.

One of the most unique features of this rule-based animation system is that it is capable of accepting the text of the dialog within the context of a database that contains facts, goals, and beliefs of the animated characters about the world and about one another. The text of the dialog is preprocessed so that the linguistic and semantic aspects of a conversation can be interpreted by computer programs in charge of the speech synthesis, semantic analysis, and generation of gestures and facial expressions.

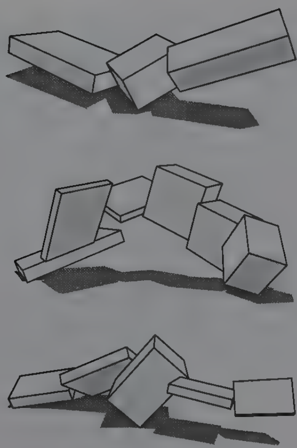
The rules of the program used to create the images in Figure 12.4.4 are based on a topology, or classification, of facial expressions and hand gestures that assigns a meaning to a selection of gestures and expressions. It also establishes a semiotic relationship and a temporal synchronization between speech, gestures, and expressions. Much of the sequencing and coordination of actions is created with a computer model that activates transitions between actions based on conditions being met, or on rules of probability.

Within the computer animation program that was used to generate Figure 12.4.4 the **gesture motion** is specified with information about the location, type, timing, and handshape of individual gestures. The hand, wrist, and arm positions can be controlled independently. An interesting feature of this system allows users to control the expressiveness of the gesturing of a character by modifying the size of its **gesture space**, which is usually dependent on the virtual age group and culture of the animated characters. The facial expressions are generated both automatically when based on the intonation and phoneme—or spoken sounds—and by hand when they add meaning to the spoken discourse. Gazing, one action involved in facial expression, is controlled automatically based on the purpose of

12.4.3 These still frames from a goal-oriented animation (top left to bottom right) illustrate how the software can generate the motion paths necessary for a character to manipulate a pair of glasses with both hands. The motion of the arms is achieved with both inverse kinematics and a sensorimotor model based on neurophysiological studies. (Images from *Planning Motions with Intentions*. Courtesy of Yoshihito Koga, Stanford University.)

HELP ME GET $50? WRITE THE CHECK

12.4.4 These gestures were automatically generated by a goal-based animation system that creates a sequence of simple actions in response to a statement. Gestures accompany requests or statements, for example, a gesture representing writing is generated from the mention of the action of writing a check. (From *Animated Conversation: Rule-Based Generation of Facial Expression*. Courtesy of Dr. Justine Cassell.)



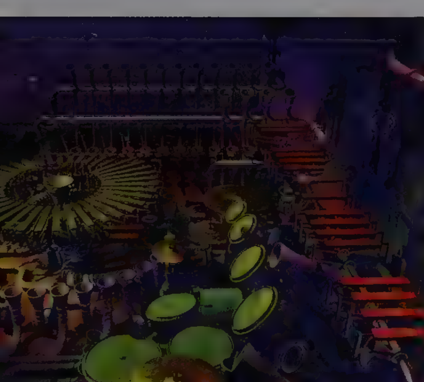
12.4.5 These simulated creatures were optimized over 100 generations for locomotion on land. Their motion techniques include shuffling, lumping, crawling, rocking, and hopping. See Figure 15.3.2 for additional detail. (Images from "Evolving Virtual Creatures" by Karl Sims. Courtesy of Karl Sims, Thinking Machines Corporation.)

looking—for example, whether to look away to gain concentration or look at the other character to reinforce a point in the conversation.

In addition to simulating human-looking characters, it is also possible to simulate and animate artificial types of life with goal-oriented computer animation techniques. The three-dimensional creatures illustrated in Figures 12.4.5 and on pages vii–ix (Table of Contents) were created with a computer program that uses genetic algorithms to define the shape of the creatures and the processes by which their motion is controlled. These creatures are the result of a genetic evolution simulation that seeks to optimize a specific goal: their locomotion on water or land. This is achieved by running survival tests throughout 100 generations of approximately 300 members each. The survival tests consisted of a physical simulation where the creatures are tested for fitness based mostly on their speed and ability to control their speed and direction of motion. The fittest creatures within a generation are selected by the program for survival and reproduction.

The creatures in this example are able to sense contact with their own selves and with the environment, and can avoid obstacles. These creatures have a morphology that evolves from a simple control system that senses the environment, evaluates the situation, and reacts to it. The evolutionary process starts with a random generation of sets of nodes (body, limbs, and head) and the connections between them. A clear set of rules and procedures governs all the stages of this simulated world, including the initial assembly of the creatures, their behavior, their fitness evaluation, and their reproduction and optimization.

12.5 Facial Animation

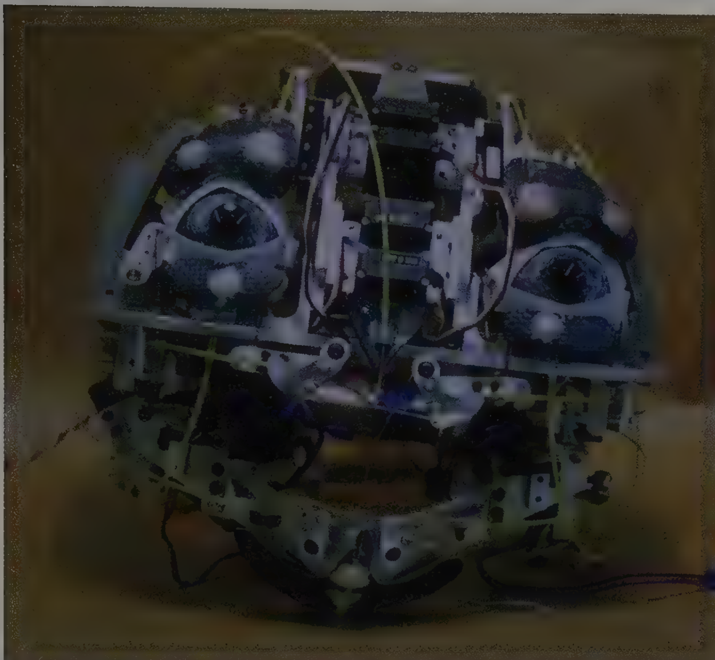


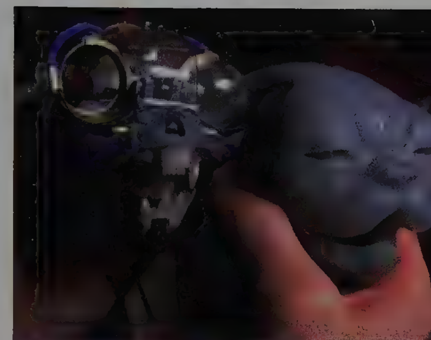
12.4.6 The virtual musical instruments in *Pipe Dream* were animated with procedural data generated in the MIDI format with MIDImotion software. (© 2002 ANIMUSIC.)

The creative desire for realistic animation has greatly contributed to the increased sophistication of facial animation controls. There are many techniques for animating **facial expressions**, including morphing between libraries of key poses, blend shapes, simulations of muscle systems, and motion capture techniques. Facial animation is often generated with hybrid combinations of these techniques, and the relationship between the inner structure and the outer surface remains a key component in the success of the final result (Fig. 12.5.1). Facial animation is usually applied to the character after the body primary motion has been blocked out (Figs. 2.7.10 and 2.7.11).

Facial Animation Tips

The facial animation of a character ends up being a big part of what audiences see on the screen. That is both because the dialogue lines that we hear are coming out of the character's mouth, and also because there is a wide range of nonverbal communication that can be achieved with facial expressions. When blocking out the animation of a face it is useful to start by animating the eyes because audi-





12.5.1 These two images illustrate the importance of animating the facial skin through the use of simulated bones, joints, and ligaments. On top, the mechanism of an animatronic cat's head built for a feature film appears alongside the skin that will eventually cover it. At left is a complex animatronic underskull for a feature film fantasy character. (Courtesy of Jim Henson's Creature Shop.)

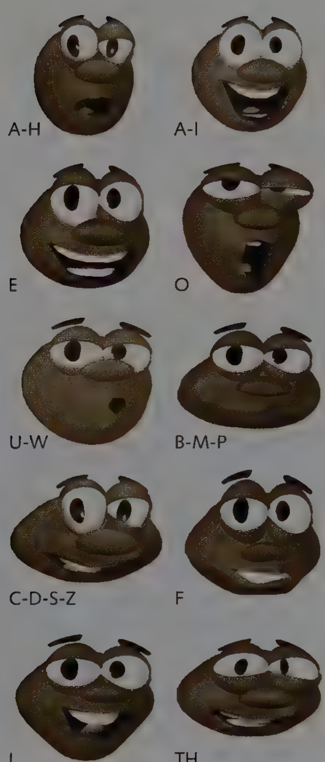
ences usually look at the eyes first. A single null point usually controls the point of interest and direction in which the eyes look. It is also important to keep different timings for each of the different major components of a face, such as the eyebrows, lips, and nose; this approach often results in more engaging gestures and overlapping secondary motion. Controlling the jiggle of skin motion, especially in characters or creatures with skin that hangs, is also an effective way to convey secondary motion and a sense of mass. When synching the lips to the dialogue it is best to focus on animating the important intonations, which usually yields better results than overanimating the lip positions. The open mouth or the closed mouth positions are the important shapes because they are the extremes that show emotion.

Morphing Targets and Phonemes

Using targets and **morph interpolation** to keyframe specific emotions and using phoneme shapes to do lip synch is a common way to do facial animation. One common side effect of using morph interpolation and blend shapes is that the resulting in-betweens become unpredictable. A **library of key expressions** is a simple and convenient way to store and retrieve many facial expressions. Libraries of key expressions are based on keyframing techniques and, therefore, require a fair amount of interactive work for placing the required expressions along the timeline. Figures 12.5.2 and 12.5.4 show a set

12.5.2 English phonemes for a cartoon character from *How to drive everybody crazy*. (© TeamTO–France 3–Cake Entertainment, 2008.)

12.5.3 Lip configurations from a library of key positions can be synchronized to a graph representing the voice soundtrack of the main character in the animated movie *Final Fantasy: The Spirits Within*. (© 2001 FFFP.)



12.5.4 Basic phonemes in the English language that can be used as target keyframes. (Sequence by Kevin Reagh, Horizons Animation (HAI). Image courtesy of the Horizons Companies.)

of simplified **phonemes** that can be used as morph targets. Figures 12.5.3 and 12.5.5 illustrate libraries of lip key positions, facial expressions, and an interface that allows animators to place those keyframes in the animation while viewing a graphic representation of the soundtrack. The **motion transitions** between facial expressions have to be checked for details in the interpolation that may look unnatural and, therefore, be distracting.

Blend Shapes

Blend shapes are used to create different expressions, or shapes, in a variety of ways (Fig. 5.7.10). **Blend shapes** can be sculpted by pulling vertices in the geometry, by using bones (Fig. 12.5.7), or by using **simulated muscles** that control both the way the skin moves and the facial expression (Fig. 12.5.6). In the latter implementation expression is achieved by tensing or relaxing the virtual muscles that control the skin position and then blended from one shape to another; the muscle forces are simulated with small virtual muscles that contract or relax (Fig. 12.5.3). This change is transmitted to a flexible lattice that represents facial skin. In order to simulate a realistic propagation of dynamic forces throughout the simulated skin, some animation systems employ a multilayer flexible lattice that permits an increased interaction of the spring forces (Figs. 12.5.6 and 12.5.8).

Motion Capture and Dynamics Simulations

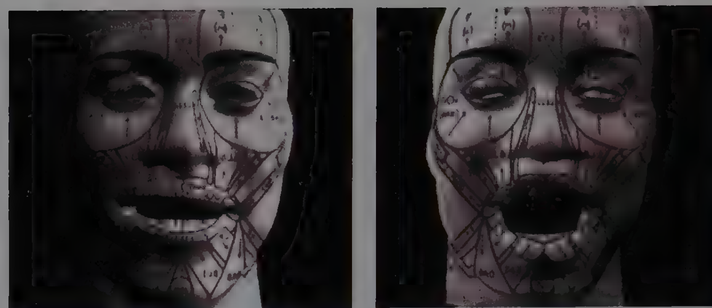
Facial motion capture data is usually obtained with a face tracker and markers on the face. The number and placement of sensors and markers in a **face tracker** varies from system to system. The model pictured in Fig. 9.3.3, for example, was captured with 80 markers and 45 control points in the Vicon system. Special contact lenses can be

placed on the eyes to capture eyeball motion. It is common to combine motion capture data with other facial animation techniques because often the motion capture data is not sufficient to generate well-defined expressions. Usually, additional animation detail is built on top of the motion capture data. Figures 12.5.11–12.5.14 show different approaches to facial animation including motion capture and blend shapes (Fig. 12.5.11), and motion capture and biomechanical simulation techniques (Fig. 12.5.12).

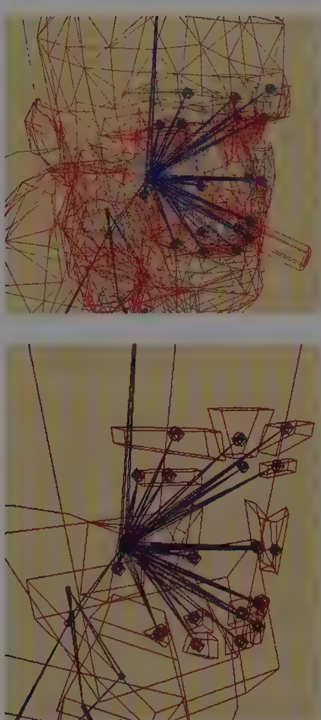
The image shown in Figure 1.4.2 was created with a multitrack facial animation system that combines simulated muscles and key expressions—called snapshots in this system—that correspond to phonemes and emotions. The snapshots for phonemes consist of the lip positions that correspond to the emission of a particular sound. Both the emotion and the phoneme snapshots can be specified with intensity parameters that control the motion of different virtual muscles. For example, a specific facial expression could be defined as: raise_sup_lip 30%, lower_inf_lip 20%, open_jaw 15%.

The emotions can also be specified in a parametric fashion by defining the change, intensity, and duration of facial expressions over time. Some parameters of emotion that can be used to control virtual muscles include the length of time that it takes for the expression to start and to decay, its overall duration, and its transition to a relaxed state. Facial animation based on goal-oriented techniques is illustrated in Figures 12.4.3 and 12.4.4.

12.5.5 Animators have a wide range of choice to dial expressions or components of expressions in this facial animation interface. (Teenage Mutant Ninja Turtles and TMNT are trademarks and copyrights of Mirage Studios, Inc. *TMNT* © 2007 Imagi Production Limited.)



12.5.6 Two views of a head model showing the relationship between simulated facial muscles and expressions. (Courtesy of Acclaim Entertainment, Inc., Advanced Technologies Group.)

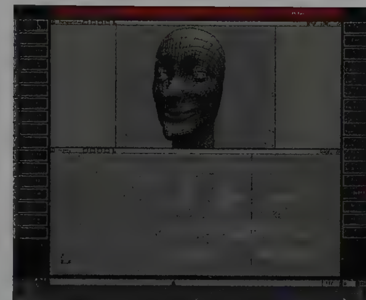


12.5.7 An example of bone-driven blend shapes. This facial animation rig for the *Medal of Honor* gunnery sergeant (with and without the polygonal skin) includes 21 bones: 8 for lips, 2 for nose, 4 for cheeks, 3 for eyebrows, and 4 for the eyes. (© 1999 Electronic Arts Inc. All rights reserved.)

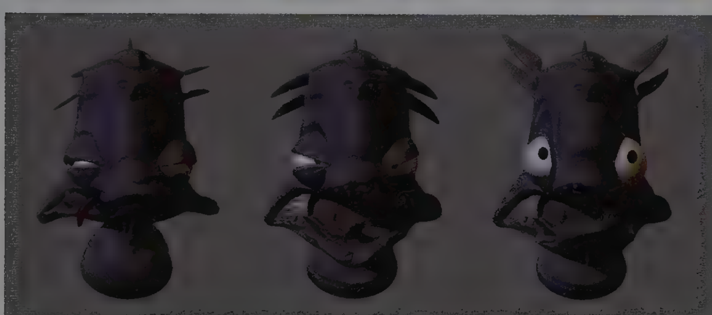
12.6 Crowd Animation

Crowds are like a large organism and, in that sense, virtual extras are not just moving props but living characters with personalities. Crowds are also, in a way, like a single actor and, in that sense, they need to be directable. Ideally, crowd animation systems combine several layers that give animators control over the actions of the entire crowd, groups within the crowd, and individual characters in the crowd. Some of the crowd animation techniques that are becoming standard tools include the ability to control the flow of moving crowds, do collision detection and avoidance, mix and match animation cycles from a library and blend between them, allow the crowd to interact with the main characters, define the eye lines, and be able to add keyframe animation to select characters on top of the library cycles. Some members of the crowd are seen only for a few seconds while others may be in the shot for several minutes, as in the bar scene in *ANTZ*. For this reason the motions of the latter must be rich and interesting.

The techniques for animating computer-animated three-dimensional crowds were greatly refined during the late 1990s, starting with feature films like *The Lion King* and *The Hunchback of Notre Dame,* and continuing with films like *Prince of Egypt, A Bug's Life, ANTZ*, and *The Phantom Menace.* In *ANTZ*, for example, much of the action is defined by the crowds. There were about 720 crowd shots with average simulations of 2,000 ants, and the largest crowd at 80,000 ants. In many of the scenes the main two characters interact with the crowd. The two were keyframed by hand, and the rest of the ants were animated with the crowd simulation system (Figs. 2.7.10 and 2.7.11). In *A Bug's Life* there were over 430 crowd shots with about 600 distinct crowd characters, and about 2,300 snippets of actions in the animation library. Each of the crowd characters was named with a personality trait (shy, funny, aggressive). In *Prince of Egypt* there were four basic models for crowd characters plus several elements created with three-dimensional morphing. The groups animated with the crowd system included the slaves, the soldiers, and the locusts. Some shots had as few as 20 and others had several hundreds of thousands of characters (Fig. 12.6.2). All crowds in this animated film were rendered with a non-realistic toon shader.



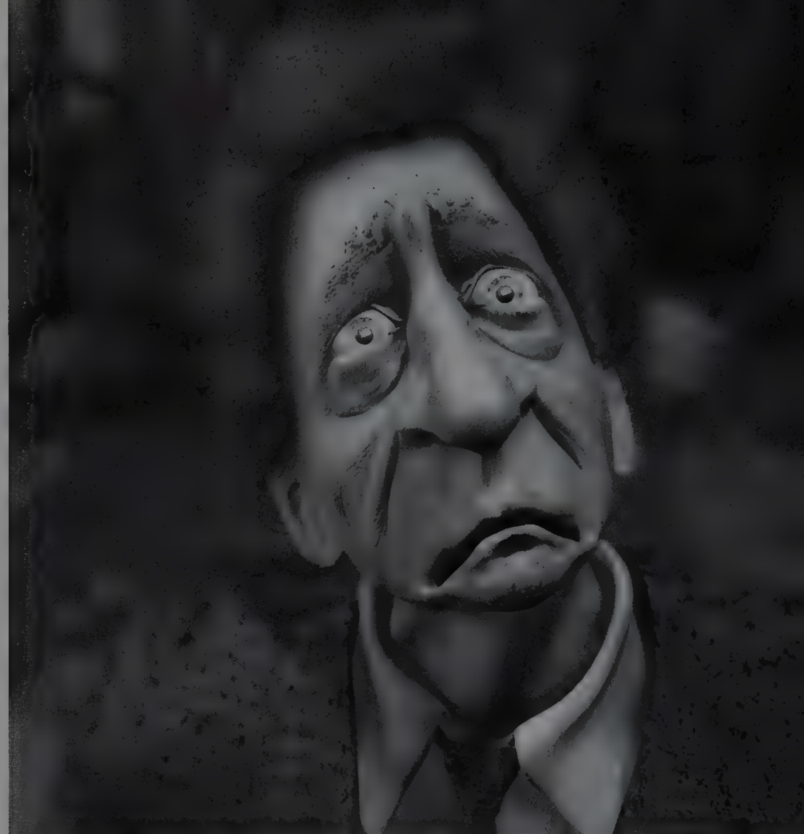
12.5.8 The blocking of the motion is done with proxy models, and it is rendered with hardware in real time. This stage of the process is used solely to judge the performance of the character. The cyan wireframe diamond at the tip of the scissors represents the point of interest of the character. Each finger has null controls and effectors (top). After the body motion is blocked, facial animation is done using the full geometry. The rest of the primary body motion is fine-tuned still with a more detailed proxy, and the secondary motion is added. Using fast hardware rendering techniques the geometry is checked and the timing is refined (middle). A detail of the final rendering is on page 293. In the animation interface (bottom) each star denotes a group of muscles that can be moved by the dashboard controls. Some muscle regions respond to more than one control. The sliders in the eye section control the eyelids, eyebrows and forehead. The lip rotation control curls the lips, and the lip push control creates a kissing motion.(Director: Daniel Robichaud, Animation Supervisor: Stéphane Couture, Art Director: Michelle Deniaud. *Tightrope*, © Digital Domain, Inc.)



DISGUST ANGER FEAR

JOY SADNESS SURPRISE

12.5.9 This character, modeled with NURBS patches, achieves his facial expressions with blend shapes. (© 2003 Oddworld Inhabitants, Inc. All rights reserved.)



12.5.10 This facial expression from *Eternal Gaze* shows the intensity and depth of the moment as the protagonist faces issues of life and death. (© Copyright Sam Chen and Aloha Animation, 2008.)

12.5.11 Expressions of wonder and smiling created with a combination of motion capture and blend shapes. (Courtesy of Giant Studios.)

In *The Phantom Menace* the battle scene between the droids and the Gungan army was created by assembling animation cycles from a library. Some of these cycles were created with motion capture and others with hand-drawn keyframe animation. The path for each creature was dictated by a particle simulator that animated a single particle for each type of creature, and then loaded the geometry and the appropriate animation cycles. Finally the software applied some generic rules, especially for the multiple physical interactions between creatures, and some customized roles. When each of the particles died so did the character that was controlled by it. The avoidance of the cavalry by the infantry was done with a collision detection system. The first episode of *Star Wars* had 110 shots with crowds, and an average of 95 frames per shot. There were about 140 animation cycles in the library, with most cycles being slightly under 200 frames long. Figure 12.6.1 shows the interface of a popular standalone software for crowd simulation, originally developed for the production of Peter Jackson's films.

12.7 Interactive Animation

The entertainment industry uses computer animation techniques in ways that go beyond the confines of a traditional movie theater or a television set. **Location-based entertainment** is another name given

12.5.12 The facial expressions of this virtual actress were created with a combination of facial motion capture and biomedical simulations of muscles and tissue. (Images courtesy of Mark Sagar, Pacific Title Mirage, 1999.)

to interactive games that are bound to a specific location due to their size and equipment requirements, virtual reality and motion rides (Figs. 12.7.1–12.7.3). The name **virtual reality** is given to applications that simulate the experience of reality based on the use computer animation and immersive technologies.

Motion Rides

Motion rides are a kind of movie rollercoasters, where computer animations are shown in a theater constructed on a motion simulator. **Motion simulators** are platforms or bases that move—usually with pneumatic mechanisms—in response to a script of programmed motion. The motion of the platform is choreographed in conjunction with the motion of the camera in the computer animation. The size of motion simulators varies, but on average they hold between ten and thirty seats. Motion rides can simulate the motion of the observer through space, or the motion of the environment around the observer, or both. Entertainment rides plunge viewers into fantasy worlds and they almost always take place in scenic theatrical environments. Rides usually explore realistic looking environments, but sometimes the environments are fantastic and have their own laws of physics. Typically the observer (the audience) rides a vehicle that can move at high speeds. This generates one of the most typical characteristics of

12.5.13 A single facial model may use facial motion capture data as a guide for deformation. Captured point data is converted to animation point data and mapped to the corresponding markers on the geometry. (© NAMCO Ltd. All rights reserved.)

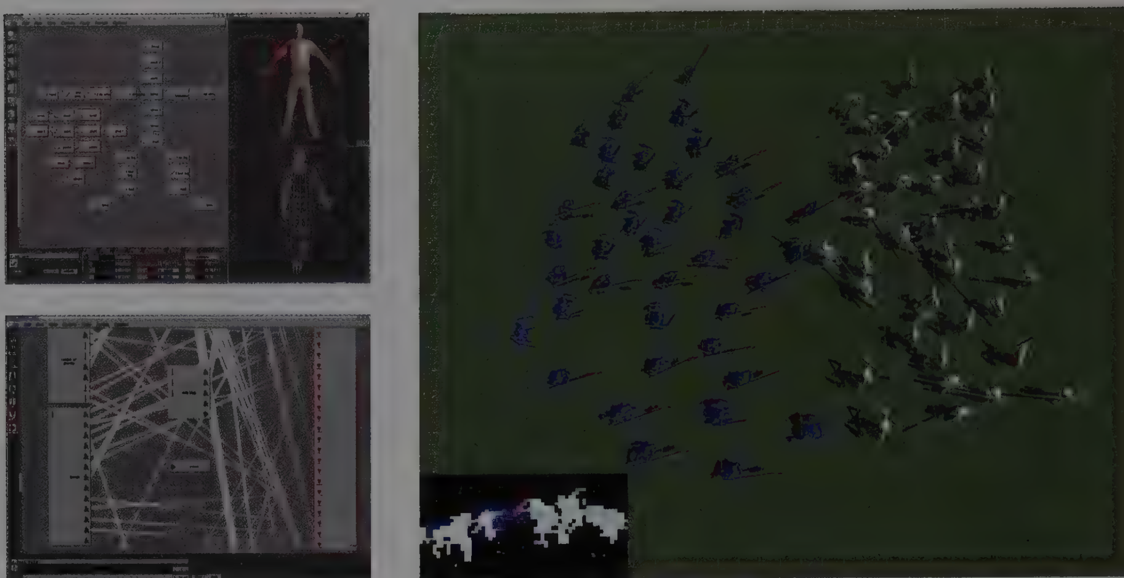
12.5.14 Eve Solal, a virtual character animated with motion capture. (© 2002 Attitude Studio.)

12.5.15 (Opposite page) A wide range of subtle emotions and states of mind can be defined with minimal changes of the facial features. Examine the different versions and combinations of surprise, worry, concern, sadness, lack of interest, annoyance, and indifference, as expressed by the protagonist in *How to drive everybody crazy*. (© TeamTO–France 3–Cake Entertainment, 2008.)

motion rides—sudden changes of speed and direction.

The **entertainment value** of most motion rides is based on the story they tell, the synchronization of the motions of the simulated camera and environment with those of the motion simulator, and the consistency of the motions and environments presented. Unlike computer animations that are meant to be *watched*, motion rides are meant to be *experienced*. Much of the effect of motion rides is based on the physical experience derived from fooling the sense of balance and orientation of the audience. This is done by simulating not only the moving images that the audience would see if they were moving, but also some of the physical sensations that they would feel if they were moving. For these reasons, the **synchronization of motions** between the motion simulator and the computer animation is essential for the success of a motion ride.

From the production point of view the synchronization of the motions is a process that requires a lot of trial and error, including the display or projection of wireframe motion tests of the computer animation while the motion of the base is fine tuned. The **field of view** and the **lines of sight** have to be carefully orchestrated so that the simulated and real motions are synchronized. An important issue to consider is the fact that the motion of the platform should be affected by both the motion of the simulated camera and some of the



12.6.1 The Massive crowd animation software (above) used for *Lord of The Rings* uses agents with programmable physical attributes that rely on fuzzy logic to respond to the environment around them. The logic nodes used to power the simulation are interconnected with the graphical interface pictured above. A library of motion capture sequences are mapped to the agents' skeletons and invoked by the software as needed. Once the agents and their variations are finalized, the crowd simulation begins on a battlefield (top right). (© 2002 Stephen Regelous.)

actions that take place in the simulated three-dimensional environment. The motion of the platform is mostly determined by camera-related issues like orientation and speed, but also by other motions—for example, an explosion or a giant monster trying to blow us away. The action in the animated environment simply drives the motions in the human environment. For example, if the simulated spaceship that carries the audience shoots at a monster in the scene, the motion platform would move based on the recoiling motion of the ship as well as the turbulent forces generated by the monster as it tried to escape.

Another factor essential to the success of a motion ride is the **consistency of the motion** in the environments simulated on the motion platform. Consistency of motion implies that the platform will always move in a consistent and similar way in anticipation, or in reaction to the actions shown in the animation. One inaccurate motion is enough to destroy the illusion of realism sought by motion rides. But consistent motion does not necessarily imply that the motion style simulates real motion. A fantastic motion ride could have its own particular laws of physics or gravity forces, for example, that are different from the natural laws of our reality. The audience can quickly adapt to any motion as long as it is consistent.

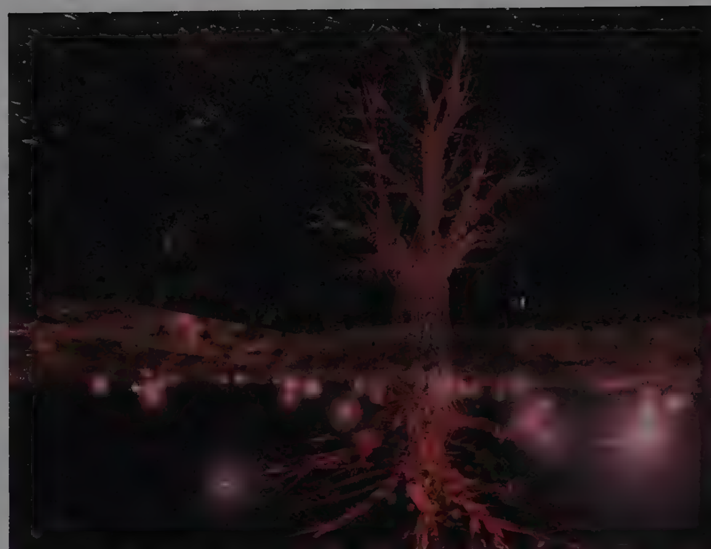
During preproduction it is important to design the animation based on the types of motions and effects that the motion simulator is capable of. It is useless to design a computer animation for a motion ride without taking into consideration the weaknesses and strengths of the motion base. The integration of computer animation and motion simulation technologies has to be seamless for motion rides to be effective. From this point of view the computer animations for motion rides are somewhat stylistically enslaved to the capabilities of motion

simulation technology. Even platforms with six degrees of freedom (rotations and translations on XYZ) have motion limitations. A debate between creators of motion rides focuses on whether the images in a ride should be stylized versions of fantastic worlds or photorealistic simulations of our world. Most agree on one issue: Regardless of what the imagery looks like, the motion should be outstanding. This is especially important in cases where the renderings are simple.

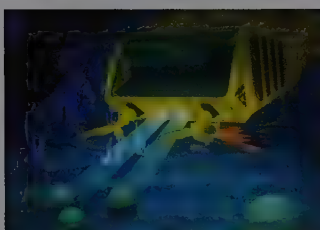
Production schedules for motion rides are usually longer than those of a computer animation intended for a traditional delivery format, due to the additional task of synchronizing the motion of the animation to the motion of the platform and vice versa. This usually requires an animation design that takes into account what the motion base can do, and also the extended production times required to fine tune the motion of the platform to the animation motion tests back and forth until the desired effects are achieved.

The film formats usually employed to record computer animations for motion rides require larger sizes and higher resolutions than other delivery formats. These formats include 70 mm film with five, eight, and fifteen perforations (also called 5, 8, and 15 perf). Both the size and resolution of the images have an impact on the computing and production times required to create animation for motion

12.6.2 This crowd scene was created from four basic character geometry models, plus several variations created with three-dimensional morphing. (Image from the motion picture *Prince of Egypt*™ © 1998 DreamWorks L.L.C., reprinted with permission by DreamWorks Animation.)



12.7.1 *Osmose*, a real-time immersive virtual environment. (Copyright Char Davies/Softimage 1995–1999.)



12.7.1 In the *Loch Ness Expedition* game, eight vehicles—each carrying six participants—are connected to a computer that generates three-dimensional computer graphics in real time. (Game developed by Iwerks Entertainment and Evans & Sutherland. Courtesy of Evans & Sutherland Computer Corporation.)

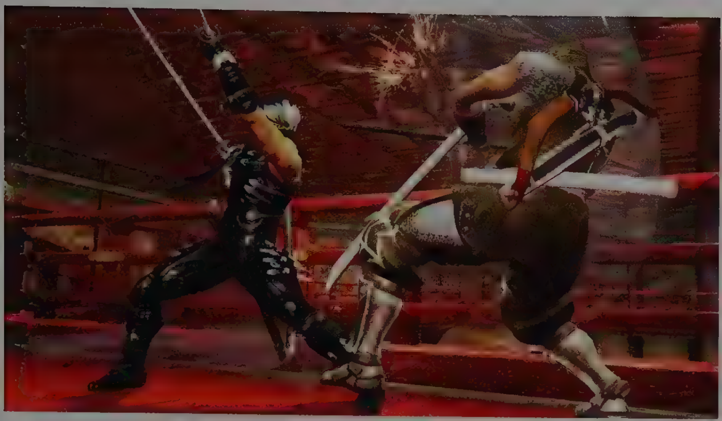
12.7.3 The *Virtual Peers* interactive system allows a child to build a Lego bridge in collaboration with two virtual children. (© 2008 Justine Cassell.)

rides. Even though motion rides typically involve a fair amount of action most of them are, in essence, passive forms of entertainment from the point of view of **audience participation**. Most forms of location-based entertainment that include interactivity are in the form of games where the audience interacts with computer animations.

Interactive Games

The functionality and image quality of today's interactive games continue to improve along with technological innovations. Today's powerful seventh generation game platforms—for example, Microsoft Xbox 360 and Sony PlayStation 3—have redefined the meaning of state-of-the-art, especially as it applies to three-dimensional computer animation. Interactive games are, in fact, so bound by technology that before dealing with creative issues it is imperative to understand the ways in which technology defines the creative boundaries in an interactive game. Early designs for an interactive game that do not consider the technical limitations of the delivery system result in wasted creative energy and costly production delays.

One of the critical aspects of many interactive games—especially action games—is the **response speed** of the game to the commands, actions, and requests of the user. This speed is determined by the processing speed of the hardware on which the game is played on, and it defines strategies for saving and displaying the three-dimensional computer animation that may be part of the game. Most **action-based games** for arcade use are delivered in self-contained kiosks with specialized hardware that is typically capable of fast response speeds. The versions of arcade action games for home use are delivered in media that plugs into game platform systems or personal computers.



12.7.4 A "third-person" camera shows the progress of a *Ninja Gaiden*™ fight as the player controls one of the contenders. (Reprinted with permission from Microsoft Corporation.)

12.7.5 (Below left) A fierce and fiery Balrog from *Kingdom Under Fire: Circle of Doom*™. (Reprinted with permission from Microsoft Corporation.)

12.7.6 (Above) The motion of the characters in the *Fable II*™ action role-playing game obeys the player's commands and the character's built-in intelligence. (Developed by Lionhead Studios. Reprinted with permission from Microsoft Corporation.)

The interactive games that incorporate three-dimensional computer animation are generally based on one of two display modes: playback of prerendered two-dimensional images, or real-time navigation and rendering of a three-dimensional scene (Figs. 12.7.4–12.7.6). The latter can be rendered in real time as the player moves through the environment including realistic effects that approach cinematic quality, especially when a fast CPU and graphics processor are available.

Action sequences that can be effective in the movie theater environment invariably require a major amount of creative and technical work to be adapted to the medium and format of interactive games. Two of the issues that always come up in these adaptations include the modeling complexity of the three-dimensional scene and the color and image resolution of the rendered images. It is often the task of creative game designers and artists to find the best compromise between playing fun, aesthetic qualities, technological capabilities, response time, modeling complexity, and image resolution—a challenging task.

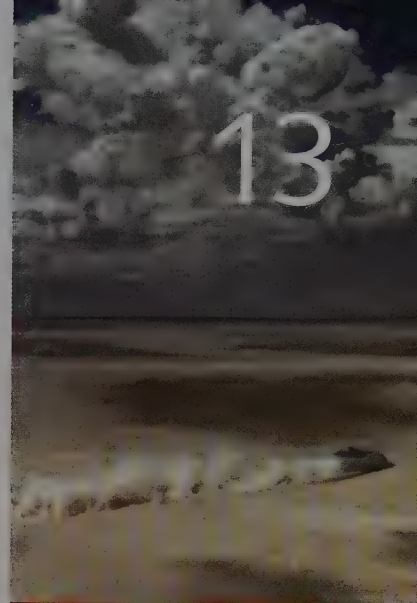
CHAPTER 12

Key Terms

Acceleration
Acoustic motion capture
Action-based games
Animation score
Attracting forces
Audience participation
Automated
Automatic collision detection
Autonomous characters
Basic tracks of motion
Blend shapes
Bounding volumes
Broken hierarchy
Center of mass
Chain root
Channel
Codified procedures
Collision detection
Collision path
Conical force
Consistency of the motion
Density
Destination mesh
Device driver
Dialog-based games
Direction
Distribution of mass
Dynamics simulation
Effector
End effector
Elasticity
Entertainment value
Environmental density
Face tracker
Facial expressions
Field of view
Filtered
Flexible lattice
Flexible objects
Fluid dynamics
Force
Friction
Function curves
Gelatin
Gesture motion
Gesture space
Global forces
Goal-oriented
Gravity
Hard rubber
Hierarchical skeleton
Hybrid environment
Impacting forces
Intensity
Intention-based
Inverse kinematics, IK
Joint
Library of key expressions
Linear force
Lines of sight
Live control
Local forces
Location-based entertainment
Magnetic motion capture
Manipulation planner
Mass
MIDI
Morph interpolation
Motion blending
Motion capture, mocap
Motion constraints
Motion dynamics
Motion planner
Motion simulations
Motion simulators
Motion transitions
Multiple layers of motion
Musical Instrument Digital Interface, MIDI
Number of motion sensors
Obstacles
Optical motion capture
Particle systems, streams
Peripheral input devices
Phonemes
Placement of the sensors
Point force
Potentiometers
Procedural-based motion
Prosthetic motion capture
Radial force
Raw motion
Real actors
Receivers
Resisting forces
Response speed
Rest position
Retargeting motion
Rigid body dynamics
Rigid objects
Rotational velocity
Rotoscoping
Rule-based motion
Sampling points
Secondary motion
Sequence of motions
Shortcuts in the simulation
Simulated muscles
Simulation domain
Source mesh
Springs
Stiffness
Strength
Stretched position
Swirling
Synchronization of motions
3D rotoscoping, 3D roto
Torques
Transfer paths
Transit paths
Transponders
Turbulence
2D rotoscoping, 2D roto
Velocity
Virtual actors
Virtual reality
Viscosity
Volume
Vortex

CHAPTER 13

Visual Effects Techniques



QUICK INDEX

(Top: Image courtesy of The Mill.)

Summary

Most visual effects today are created with a combination of digital and traditional techniques. Often the creative and production approach to creating visual effects for a live-action project is slightly different from a computer-generated character animation project, and some of those differences are discussed in this chapter. Most complex visual effects shots require the compositing techniques explained in Chapter 14 to integrate the multiple layers created with different effects techniques. Fortunately most of the digital and animation techniques required to create high-quality visual effects work can be implemented today with desktop computer systems.

13.1 Basic Concepts of Digital Visual Effects

Not too long ago visual effects, or **VFX**, were prominent only in so called effects-driven live action feature movies (Fig. 13.12.3). But in recent years the use of visual effects has expanded into mainstream movies where a few visual effects shots play a supporting but significant role in the storytelling. Visual effects have also grown significantly in other areas such as TV series and platform games. The process of creating visual effects for a feature movie, TV commercial, or music video starts well before the production of the final images. As mentioned in Chapter 2, preproduction and planning are essential in any project that involves computer animation, but this is particularly true in the production of visual effects for live-action. The increased need for planning is partly due to the increased number of processes and people involved: lighting the set, travelling to remote locations, changing weather, camera crews and live actors, to name a few. Few of these factors impact the average character computer animation project. The creative demand for constant innovation and the changing nature of the techniques and technologies used to create visual effects make it a necessity to start visual effects preproduction early in the planning process. Visual effects projects demand custom solutions (Fig. 13.1.1).

13.1.1 How many visual effects techniques can you recognize in this shot from *Spy Kids?* (© 2002 Hybride. Images courtesy of Dimension Films.)

Generally speaking, all visual effects fall into one of four major families of effects, but there are a few **cross-over techniques**. Some effects are about *matching* the live-action source—for example, the camera tracking techniques. Many effects are about *combining* elements from multiple sources—for example, the compositing techniques. Other techniques are about *adding* or *deleting* elements, like set extensions, or about *transforming* the source image—for example, color correction.

The Core Visual Effects Team

Depending on the complexity of the shots, budget and delivery schedule, the visual effects crew may range from half a dozen individuals to a hundred or more. But there are a few key roles that, no matter how large or small the project might be, are always present in any visual effects team. The exact job titles for each one in the **visual effects core team** are sometimes different between companies but the roles performed are about the same. The tasks in a big-budget production are typically divided among several individuals, while a single individual in a low-budget production might have multiple roles and responsibilities (Fig. 13.1.3). Large facilities often seek **specialists** while small studios prefer a higher ratio of **generalists**. A visual effects artist or animator in a low-budget production might have limited resources available to get the job done. But the same artist might also have flexibility to cross-over between different areas of expertise and participate in different aspects of the project. A digital artist in a big-budget show typically has extensive resources but also must focus on a single specialized task for the duration of the project. Often we fantasize about how wonderful it would be having the best of both worlds: the flexibility of a low-budget project with the plentiful resources of a big-budget show.

13.1.2 This still frame from aptly titled *Avenging Fist* illustrates a trend of low-cost and high-quality visual effects, and is also an example of the character extension technique. (© StarEast/Bob. Images courtesy of Menfond.)

A VFX team is put together based on the specific needs of a show. Generally speaking the key members in a core VFX team for a live-action show include a producer, a supervisor, some leads (or additional supervisors depending on project size) and a few specialists. The **visual effects supervisor** is assigned with planning the overall production of visual effects and is also responsible for the final quality of each and every shot. This individual usually attends the preproduction meetings that involve visual effects. He or she supervises the group heads in the team, and interacts directly with the **director** of the movie and the **cinematographer**. In projects that involve multiple visual effects production companies a senior VFX supervisor makes the rounds reviewing work and providing feedback and direction. The **visual effects producer** is responsible for making sure that the crew has the necessary resources to get the job done within budget and that the team delivers the expected effects shots within the budget and deadline. This individual is also responsible for dealing with most of the legal and talent issues such as contracts, hiring negotiations, performance reviews, and space assignments. The VFX producer must constantly balance expenses and resources with complexity and quality, and frequently deals with rescheduling due to script revisions, production delays, and securing rendering resources. The **sequence supervisor**, the **computer animation supervisor**, the **compositing supervisor**, and other department supervisors are responsible for directly overseeing the work of the members in their respective areas. See Chapter 2 for additional descriptions of roles within small and large computer animation projects.

The Visual Effects Pipeline

Planning and creating visual effects follows a **production pipeline** with multiple steps, many of which are described in Chapter 2 (Fig.

Pirates of the Caribbean

Visual Effects and Animation ...381
Industrial Light & Magic:
Digital Production Supervisor 1
Compositing Supervisor 1
TD Supervisor 1
Creature Development Supervisor.1
Digital Model Supervisors 3
Additional VFX Supervisors 2
Visual Effects Art Director 1
Visual Effects Assoc. Producer.......1
Assoc. Animation Supervisor.........1
Sequence Supervisors.................22
Animators 49
Digital Artists 161
Digital Models............................27
Visual Effects Editor......................1
Lead Location Data Capture 1
Senior Visual Effects Coordinators.3
Visual Effects Coordinators...........3
Model and Miniatures Unit Sups...5
Model and Miniature Unit 25
Research and Development 10
Production and Tech Support......10
ILM Senior Staff...........................4
Additional VFX Supervisor............1
Asylum:
Senior Visual Effects Supervisor.....1
Compositing Supervisors..............2
Visual Effects Producer.................1
Visual Effects Coordinator.............1
Compositors.................................5
Rotoscope/Paint Supervisor 1
Rotoscope/Paint Artists................2
CG Supervisor..............................1
Lighting 1
Matte Painting.............................1
The Orphanage Inc.:
Visual Effects Supervisor...............1
Visual Effects Producer.................1
Digital Production Manager..........1
Computer Graphics Supervisor......1
Digital Artists 2
Compositor..................................1
CIS Hollywood:
Visual Effects Supervisor...............1
Visual Effects Producer.................1
Visual Effects Production Mgr.......3
Digital Compositing Supervisor 1
Color and Lighting Supervisor.......1
Compositors.................................4
CG Animator................................1
Pacific Title and Art Studio:
Visual Effects Supervisor...............1
Executive Producer.......................1
Digital Coordinator 3
Inferno Compositors 2
Digital Compositors.......................5

Directing/Supervising................37
Director.......................................1
Writers..2
Characters Created by 4
Producer 1
Executive Producers 4
Director of Photography 1
Production Designer......................1
Editor..1
Costume Designer.........................1
Visual Effects Supervisor...............1
Music Composer...........................1
Music Supervisor...........................1
Casting 2
Executive in Charge of Production 1
Creature Concept Designer...........1
Conceptual Consultant..................1
First Assistant Directors.................2
Second Assistant Directors 2
Film Animation Supervisor 1
Associate Producer........................1
Production Supervisor 1
Production Controller...................1
Script Supervisor 1
Supervising Art Director 1
ILM Animation Supervisor.............1
ILM Visual Effects Producers 2

Art Department..........................37
Art Directors 3
Assistant Art Directors...................6
Set Decorator...............................1
Construction Coordinator..............1
Set Designers 9
Props Set Designer........................1
Conceptual Artists.........................3
Illustrators 5
Model Makers/Sculptors 2
Model Maker...............................1
Graphic Designer 1
Art Department Administrator 1
Researcher 1
Second Art Dept Administrators....2

Editing......................................28
Visual Effects Editor......................1
First Assistant Avid Editor..............1
Additional Film Editor....................1
Assistant Editors 2
Apprentice Editors.........................2
Post Production Supervisor 1
Post Production Coordinators........2
Post Production Assistants 2
Supervising Sound Editor/Designer1
Supervising Sound Editor 1
Sound Mixers...............................2
Sound Effects Editors.....................6
Supervising Dialog Editor 1
Dialog Editors...............................3
Supervising ADR Editor.................1
ADR Editors 1

Camera......................................25
Camera Operators/Steadicam 3
First Assistant Camera...................4
Second Assistant Camera 5
Film Loader..................................1
Camera Department Assistants 2
Aerial Coordinator........................1
Aerial Unit Dir. of Photography.....1
Aerial First Assistant Camera.........1
Underwater Dir. of Photography...1
Underwater First Asst. Camera......3
Underwater Second Asst. Camera.1
Camera Head Technician..............1
Camera Technician.......................1

Grips, Electricians, Rigging.......120
Stunts 91
Construction..............................89
Additional Production.................73
Cast 63
Costume 56
Make-Up and Prosthetics...........54
Transportation 50
Property and Set Dressing..........49
Music 46
Marine Team 35
Sound 31
Hairstyling.................................29
Accounting 13
Digital Intermediate....................11
Animal Training 11
Special Thanks 10
Previsualization 6
Casting and Extras Casting...........6

Caribbean Unit 168
Second Unit...............................40

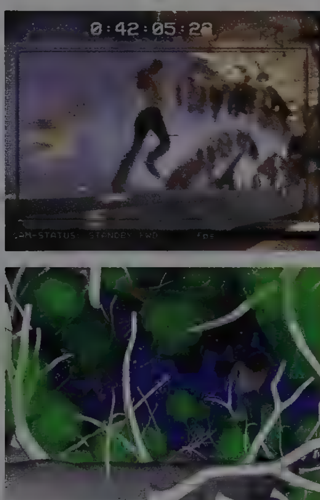
13.1.3 Partial listing (by size of group) of screen credits and statistics of most of the major departments involved in the production of *Pirates of the Caribbean: At World's End*, a classic visual effects live-action Hollywood blockbuster.

2.7.3). Concepts and stories are developed during preproduction, usually with the aid of storyboards, animatics, and previsualization (Figs. 13.1.5 and 13.1.6). The final, or almost final, script is read by the VFX producer and supervisor who determine together the type of visual effects required. The shots with visual effects are "broken down," or analyzed into the specific techniques that should yield the desired result. Once the effects shots have been broken down the work is split into practical, or physical, and digital effects, and assigned to one or several production companies. Media delivery requirements are also locked down at this stage in the process.

Previsualization, or **previz**, and three-dimensional animatics are a crucial component of the VFX production process because they allow the director, cinematographer, and VFX supervisor to conceptualize and refine a shot, and also to plan the actual shooting (Figs. 13.1.5 and 13.1.6). Most visual effects shots are the result of multiple VFX techniques used in combination. Consider, for example, the following effects **shot breakdown**: a live-action shot that involves a motorcycle stunt and pyrotechnic explosions shot in real time with a moving HD video camera against a blue screen. After converting the video into RGB image files the wires holding the stuntman are

13.1.4 A finished "money shot" in a big-budget effects feature film, such as *The Lord of the Rings: The Two Towers*, is likely to use many of the bells and whistles in the visual effects toolbox. (© MMII, New Line Productions, Inc.™ The Saul Zaentz Company d/b/a Tolkien Enterprises under license to New Line Productions, Inc. All rights reserved. Photo appears courtesy of New Line Productions, Inc.)

13.1.5 The computer animation for a Levi's commercial was previsualized within the layout of the real and virtual sets (right). The live-action scenes shot in a blue screen stage (below) were used to previsualize the animation and layout of the shot, and to do a preliminary composite. See related Figures 13.3.1 and 13.6.1. (Images courtesy of Framestore CFC and Bartle Bogle Hegarty.)



PREVISUALIZATION LAYOUT

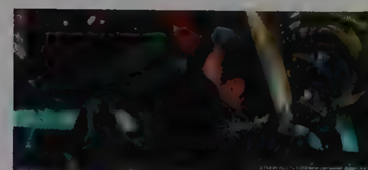
PRELIMINARY COMPOSITE

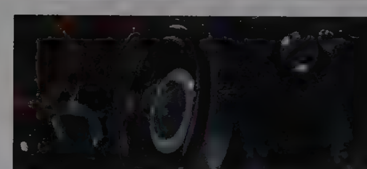
removed from the images. The digital frames are also tracked so that additional smoke trails can be added with computer-generated particles, and so that the set extensions can be matched to the live-action movie. This type of **effects salad** approach is common in effects movies, and is also increasingly used in well-organized low-budget productions that require sophisticated visual effects (Fig. 13.1.2).

With the exception of a few specialized visual effects techniques, such as camera tracking for example, the production of three-dimensional computer animation for visual effects is quite similar to the process followed for creating an all-computer-animated movie. This process, described in detail in Chapter 2, includes the major stages of preproduction, modeling, rigging, animation, texturing, and rendering (Fig. 2.7.3). The need to **match reality**, and the dependence and key interaction with the live-action crew are two of the main differences between a character animated movie and a VFX movie: a radical difference in creative philosophy and many practical implications. A top priority in a VFX movie is that the visual effects look real at all levels (Fig. 13.1.4). A character animated movie may require exaggerated performances, squash and stretch for example, non-photorealistic rendering, and anatomically impossible poses (Fig. 10.4.10). But visual effects shots must fool the eye of moviegoers and make them believe that what they are seeing is real and not a manufactured moment. Nowhere is the interaction with the live-action material more evident than during the compositing process. While compositing is something of a convenience in an all-computer-animated movie, it functions as a fundamental condition in the production of a VFX live-action movie. Compositing is used throughout the process: first to deliver temporary animatics or placeholder effects, called **temps**, later to incorporate changes and revisions, and finally to deliver the completed shot. As explained in Chapter 2, workflows are constantly fine-tuned and adapted to the needs of the project. Reviews of completed work and making revisions and changes are integral aspects of the VFX pipeline.

On the Set

Visual effects supervisors spend a considerable amount of time on the set, in the film laboratory, and at the postproduction facility.

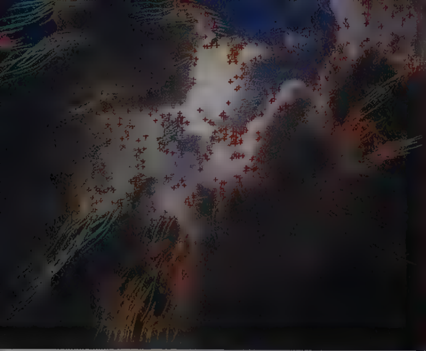




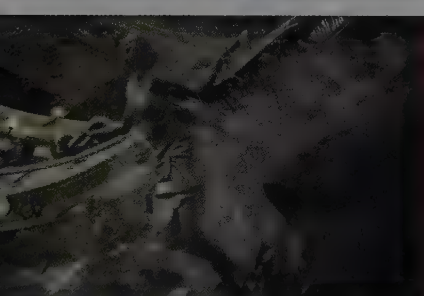
13.1.6 Most visual effects films are previsualized extensively, helping the process of breaking down the shots and refining the storytelling. Notice the close match between previz and final shot, and the lens and action notations in these previz for *Iron Man*. (Images courtesy of Pixel Liberation Front. © 2008 MVLFFLLC™ and © 2008 Marvel Entertainment. All rights reserved.)

Much of what ends up in a VFX shot starts on the set where the **cinematographer** and his/her team record the live-action material. They light the scene, measure the light, and establish the recording parameters (Fig. 13.12.5). The main task of the visual effects supervisor on the live set consists of making sure that the capture of data, images included, is done taking into account the needs of the VFX production process. Large movie productions may have multiple units of photography and one of them, usually the **second unit of photography**, is in charge or recording the live-action **background plates** (full frames) and the **VFX elements** (elements to be cut and pasted) that will become the sources for the visual effects work.

Deciding whether to use film or video to record the action is usually the decision of the director and cinematographer, also called **DP** or **DOP** for director of photography. Traditionally film negative was the only alternative for recording high-quality movie material, but high definition (HD) video is also used. *Star Wars: Episode II* and *Spy Kids 2* (Figs. 13.1.1 and 8.6.4) were two of the first high-profile movies with massive amounts of visual effects that employed HD video as the



13.2.1 Two-dimensional motion tracks created with boujou tracking software after analyzing several frames of live-action moving clouds.(© 2002 2d3 Ltd.)



13.2.2a Inexplicable things happen during this *Nintendo's Game Boy Advance* commercial: one of the Caryatides sculptures that decorates the theater comes alive and the chandelier transforms itself into a dragon. The dragon chandelier was over a million polygons, and was modeled with scripts to generate 45,000 crystal pearls on NURBS surfaces that were deformed and animated around the main dragon skeleton.

recording medium. Both of these productions used the blue/green screen technique throughout the show, and that process usually involves the participation of a VFX supervisor. Another key responsibility of the VFX team on the set is to make sure that sequences that might involve any kind of removal are shot twice: once with actors and once without. The latter is called a **clean background plate**.

In addition to the visual information recorded on film or video, other types of captured data include position markers to aid the camera tracking process, motion capture information, camera reports, survey data, and still images that can be used for texture maps. The **tracking markers** are essential to aiding the camera tracking process. These markers are made of different materials including reflective tape or small plastic spheres, including tennis and ping pong balls, and they must be placed within the field of vision in areas that might be easy to retouch later on in order to remove the markers. A locked-off camera does not require any tracking markers, a simple camera moves might require a small amount of markers, and a multi-rotation move might require several. **Lighting markers** are used to record the intensity and color of light, and the position of light sources. These markers are usually spheres, between one and two feet in diameter, each mounted on a rod so that they can be positioned anywhere on the set. The spheres are recorded at the beginning of a sequence as cameras roll and record the slate. One type of spherical lighting markers is reflective and is mostly used to capture environment reflections. Another is matte white and used to measure light color temperature, position of light sources, and rate of decay of spotlights. The third type of lighting marker is painted with a 17.5 percent gray color to use as a neutral average reflectance reference.

Camera and survey reports are helpful to recreate digitally the camera and environment. **Camera reports** include the lens focal length (in millimeters) and aperture (f/stop), as well as positional and rotational information (distance to subject, height, and tilt, pan, and roll angles). Some digital cameras and some motion control systems are able to capture this and other types of **metadata** automatically and record it onto videotape or other type of magnetic media. **Survey data** include ambient and incident light readings, and the position of markers and their distance and angle from the camera. When required survey laser systems can be used to calculate distances and angles or to scan portions of the set and convert it to XYZ modeling information (Fig. 5.6.13).

The capture of still images that can be used for mapping can be done with a 35 mm still camera or with a **digital still camera** of 4 **megapixels** (about 2,500 × 1,600 pixels of resolution) or, ideally, 8 megapixels (about 4,000 × 2,000 pixels). It is important to capture these image maps with as much color depth as possible, ideally 10-bits per RGB channel or above, although 8-bit color is more common in consumer still digital cameras. These image maps can also be used for high dynamic range rendering. (Read Chapter 6 for more information on HDR rendering, and Chapter 15 for color depth.)

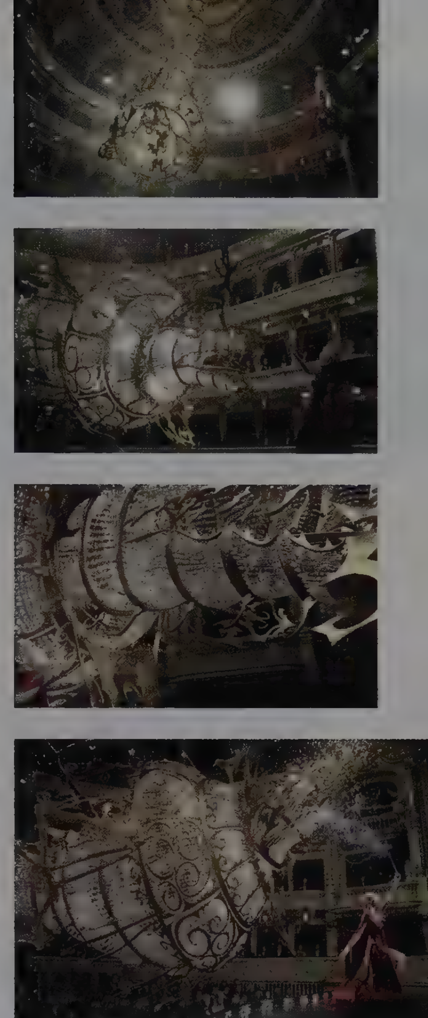
Scanning Film and Delivery

After the background plates are recorded onto film or video they are brought to the digital environment so that effects can be added. For standard feature movie work film is usually scanned, or digitized, at 2K pixel resolution and 10-bit color. 4K **film scanning** is also used but is not yet practical for all productions due to the huge file sizes and increased cost of processing. A **2K file** can be defined in a variety of ways, but it includes 1920 × 1080, and 2048 × 1152 pixels. For low-budget or TV delivery 1K scans (about 1,280 × 720) usually provide sufficient quality since the resolution of standard television is around 525 horizontal lines. (Read Chapter 15 for additional details on pixel and color resolution, and aspect ratios.) High definition video is routinely converted to RGB files of 1920 × 1080 pixel resolution and color resolution ranging from 8-bit to 10-bit depth, and 3:1:1 to 4:4:4 sampling. As a general rule in VFX production, more image resolution is better than less. This means that the greater pixel count and color depth the better results we are likely to get after the image has been retouched, composited, and color graded a few times. See Figures 14.1.2, 14.1.4, and 15.2.1 for examples of image quality loss due to insufficient resolution.

When the VFX shots are completed and approved they can be output to film and video, following some of the resolution considerations just mentioned. (Read Chapter 15 for a detailed description of output). The VFX shots are then delivered and cut into the movie, which at this point is ready to go through the color grading process. During this stage the director of photography supervises color adjustments in specific scenes or throughout the entire movie. This is usually done right before the final mastering and release to audiences in movie theaters, television, or DVD. Color grading in itself is not a visual effects technique, but it deserves mention here because it is so commonly used to fine-tune and twist the mood and the look of the finished images (Fig. 14.5.1). Sometimes background plates are color graded *before* they are given to the VFX team. While not common, this is usually done when the original plates require drastic color shifts that might impact the final image quality if applied after the VFX are added. This can also be done to save time later in the process.

13.2 Camera Tracking

Matching live-action with computer animation is easy as long as there is a **locked off camera**, one that does not move. But as soon as the live-action camera moves we need a way to keep track of that motion. That way we can match our virtual camera with the real camera and give the illusion that the live-action plate and the computer-generated elements exist all in the same time and space. The motion of a real camera and scene elements and their virtual counterparts can be matched with camera tracking and motion control. Camera tracking has become one of today's key visual effects tech-



13.2.2b The dragon was ray-traced to refract the environment through the crystals, and the multiple layers with the dragon renderings and mattes were composited into the live action. The gorgon creature was modeled with polygons, smoothed with subdivision surfaces, and keyframe animated. Much of the geometry and image maps were extracted from photographs using 3dEqualizer camera tracking software. Animation: Softimage XSI, compositing: Discreet Inferno. (*Symphony* images courtesy of La Maison. © 2002 Leo Burnett - Quad - B. Aveillan. All digital visual effects done by La Maison.)

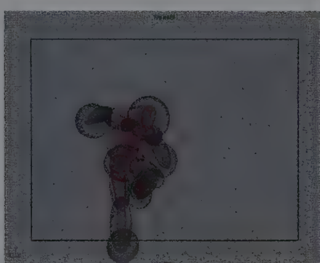
niques because it allows you to match a computer-generated object or camera with the motion, speed, and acceleration of a real camera *after* the live-action has been recorded. The **camera tracking** technique in itself does not produce finished images but numerical information that is used to match the movement of computer animated props, characters, or camera with the live-action camera (Fig. 13.2.1). Camera tracking, or **match move** as it is also called, is done by placing several markers on a few significant still frames of the live-action so that the software can automatically identify the new position of the markers on all frames of the shot. After all the frames have been tracked a **motion path** is generated and used to match the computer-animated camera and objects to the live-action footage (Fig. 13.2.2).

Camera stabilization is a technique often used in conjunction with camera tracking. By comparing the positions of the same reference point in contiguous frames this technique can determine the amount of camera jitter and smooth out, or stabilize, the motion differences above a certain threshold between frames.

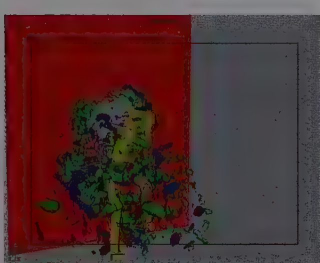
13.3 Rotoscoping

The technique of **rotoscoping** is a versatile way of matching animated elements to live-action footage. Originally developed by the Fleischer Brothers in the 1915, this technique was used by animators to match their work to live action. They would manually trace live-action frames to find points of reference for the animated cartoons. A few decades later, in the years of optical compositing and handmade matte paintings, rotoscoping was used to create moving or traveling mattes by tracing the moving elements of still frames one frame at a time. Those tracings can be used as masks with which three-dimensional objects can be laid over a live-action sequence. The basis of this technique is still the same today even though some of the rotoscoping processes can now be automated and the composting environment is almost exclusively digital. A newer form of rotoscoping is used to approximate the position and motion of three-dimensional computer animated models to live action. This form of **three-dimensional rotoscoping** uses the live-action frames as a guideline to adjust the position of three-dimensional models in the desired way. The example in Figure 13.3.1 uses the spatial and motion information extracted through three-dimensional rotoscoping to simulate the collision between the running live-action actor and a wall. The particle animation driven by the rotoscoping information and a finished composited frame is shown in Figure 13.6.1.

3D ROTOSCOPING



COLLISION VOLUMES



ANIMATION

13.3.1 Three-dimensional rotoscoping was used to create the *Odyssey* Levi's commercial. The video image of the actress is rotoscoped with a simplified three-dimensional reference model (top). The proxy model is used to establish collision fields (middle) that are used to drive the particle animation of wall debris and dust shown in Figure 13.6.1. (Images courtesy of Framestore CFC and Bartle Bogle Hegarty.)

13.4 Blue and Green Screens

Both blue and green colored screens are used to extract traveling mattes for live-action shots. The technical name for this technique is **chroma key** since any color can be used to achieve the effect of inserting, or keying, a foreground image over a flat color background.

13.4.1 The blue screen truck was composited over a digital matte painting that combines mountains from two different locations. The bridge combines the logs from the blue screen shot with the supporting underconstruction of a 1-to-15-scale bridge miniature. A shake was added to the camera during compositing to simulate that the scene was shot from a helicopter. (*Coronado* images courtesy of Uncharted Territory, LLC.)

13.4.2 An exterior blue screen live action scene shot with a Sony CineAlta 900 HD camera was composited with After Effects software over a cave and pyramid miniature and computer-generated vehicles in the background. (*Coronado* images courtesy of Uncharted Territory, LLC.)

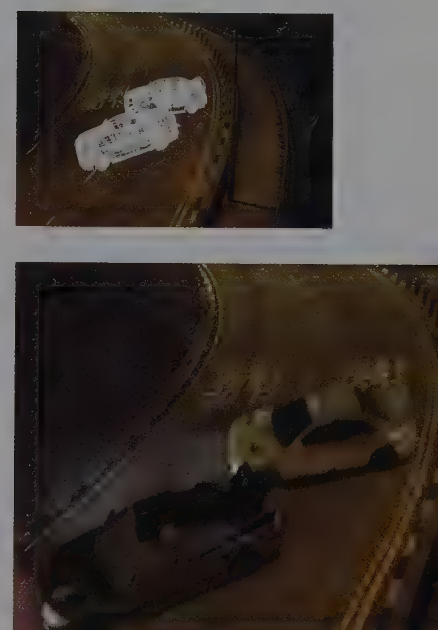
13.4.3 A straightforward digital matte painting shot, where some foreground elements were removed, and multiple stills were assembled into a new panoramic background. (*Coronado* images courtesy of Uncharted Territory, LLC.)

13.4.4 The movie *L'Anglaise et le Duc* by director Eric Rohmer was constructed from matte paintings and replications of crowds shot in green screen stages. (Visual effects by Buff. © Pathé, CER.)

Historically the **blue screen** was more common in film production while green screens were used in television production, but today they both yield comparable results. However, when shooting on film stock with high ASA ratings blue screens can be problematic since the blue emulsion layer in fast film is the one with the most visible grain. Another key factor in choosing either a blue or a green screen has to do with the colors of the foreground elements in the shot. Blue is a convenient color for matting because the flesh tones of most actors do not have any blue. Blonde hair, for example, has a fair amount of green. If the actors were wearing blue clothing though, **green screen** would be a better choice to pull mattes off the flat background.

With blue or green screens the actor or model to be matted over a background is positioned in between the camera and the screen. Since the background is of a uniform color, it is easy to isolate it from the rest of the colors in the scene and eliminate it from the negative (Figs. 13.2.4, 13.2.5, 13.4.1–13.4.4, and 14.3.7). The color background, blue or green, can also be made black by using color filtering and color correction techniques. Blue screen shots yield a monochromatic matte composed of a flat background and the flat silhouettes of foreground objects or actors. Lighting blue or green screen shots is always done to avoid **color spill**, which is light reflecting blue or green off the screen and onto the actors. The final matte is used to composite, or combine, the color image of the background and the color image of the foreground. Read Chapter 14 for additional technical details on the compositing process.

13.5.1 The number of crates in this outdoor storage facility was increased using set extension techniques. The scene was recorded with the Cinemascope, 2.35:1, aspect ratio. (*Gorgeous* images courtesy of Menfond. © GH Pictures China Ltd.)



13.5.2 The cars in *Legend of Speed* are computer-generated props rendered with motion blur and volumetric lights. (Images courtesy of Menfond. © China Star Entertainment Ltd. / Win's Entertainment Ltd.)

Background Replacement and Wire Removal

Background replacement and wire removal are two techniques that go hand-in-hand with chroma key and blue/green screen techniques. **Background replacement** is achieved by compositing a new background into the flat color area, these can be computer-generated or matte paintings painted by hand with traditional or digital tools (Figs. 13.4.1–13.4.4). For shots that do not use blue or green screen and that have moving cameras, it is necessary to use either camera tracking or motion control for creating the traveling matte to insert the new background. **Wire removal** is a simple technique that starts by

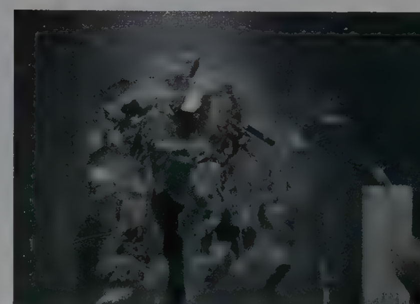
13.5.3 Before-and-after adding the eye digital props in *Ultraviolet*, shot on HD, and produced with a Maya-RenderMan-After Effects pipeline. (Images courtesy of Menfond Electronic Art & Computer Design Co. Ltd.)

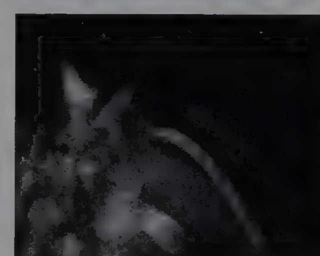
13.5.4 Before-and-after still frames from *Battle of Wits*. Notice the red marker on the actor's hand to facilitate camera tracking and placement of the virtual arrow. (Images courtesy of Menfond Electronic Art & Computer Design Co. Ltd.)

13.5.5 (below) The performance of an actor hanging from wires was recorded against a green screen. The orange markers on his back were used to match the computer-animated wing extensions to the position and movement of the character. (Courtesy of Menfond. © GH Pictures China Ltd.)

13.5.6 A virtual stunt double of the real actor was built to facilitate the animation and integration of digital character extensions, shown in Figure 2.7.6. (Courtesy of Links DigiWorks Inc. © 2005 SHINOBI Film Partners.)

13.6.1 (Below) The particle animation of wall debris and dust was driven by a rotoscoped proxy model and the collision fields shown in Figure 13.3.1. (Images courtesy of Framestore CFC and Bartle Bogle Hegarty.)





DUST PARTICLES

WALL DEBRIS

identifying in the background plates the wires and rigs that hold actors and props or the markers used for camera tracking. Wires or undesired images on the plate can be replaced with matching sections from a clean background plate or by hand-painting over them. Figure 13.5.5 shows wires and markers removed, and Figure 1.5.4 shows still frames from a martial arts movie with the wires removed.

13.5 Set and Character Extensions

Often it is easier, faster, and sometimes cheaper to build a set or a portion of it with three-dimensional computer-generation techniques than with real materials. **Set extensions** are usually built from measurements taken on location or by extracting the dimensions and camera points of view from the background plates. Many techniques can be used to build virtual set extensions including photogrammetry, covered later in this chapter. But the most common approach to computer-generated set extensions is modeling the three-dimensional geometry using the background plates as reference (Fig. 13.5.1). The live-action productions shot entirely on green screen require that all the sets, not just a few extensions, are built virtually and composited with the live-action performances (Fig. 13.12.5). **Character extensions** can be built in a similar way, but because characters tend to move more than sets it is fairly common to use rotoscoping for tracking the extension to the motion of the character (Figs. 11.2.6, 13.1.2,

13.6.2 Particles were used extensively to create explosions in this helicopter attack shot, where the sky is the only live-action element. *Coronado* had 614 visual effects shots that were completed by a compact team of 15 digital artists, animators, and technical directors. (Courtesy of Uncharted Territory, LLC.)

13.6.3 Stills from the dynamic simulation of a crash. The colors in the wireframe version represent the transition of particles from fire to smoke. The rendered frame includes the dust clusters created as debris from the crash hits the ground. (© CA Scanline/creaTV/Pro7.)

13.5.3–13.5.6). **Computer-generated props** are also popular these days and they fall somewhere in between set and character extensions. Static props are similar to set extensions while props that live actors interact with are closer to character extensions (Fig. 13.5.2).

13.6 Computer-Generated Particles

Computer-generated particles is a versatile technique that has become a staple in visual effects production, and it is also the bread-and-butter of many effects production houses. This technique essentially simulates the motion of particles that are subject to a variety of forces (Figs. 13.6.1 and 13.6.2). By rendering thousands or millions of these particles in a variety of ways it is possible to simulate different materials such as fire, smoke, liquids, soil, and even hair. Read Chapter 12 for more information on motion dynamics and procedural techniques used to animate particles (Figs. 12.3.4 and 12.4.1). Figure 13.6.3 shows a realistic rendering created with particle systems to simulate an airplane crash, the simulation is calculated in several passes: first the debris alone, then the debris, fire, and smoke. The soccer images on page 426 show a stylized effect including particles and other techniques.

13.7 Crowd Replication

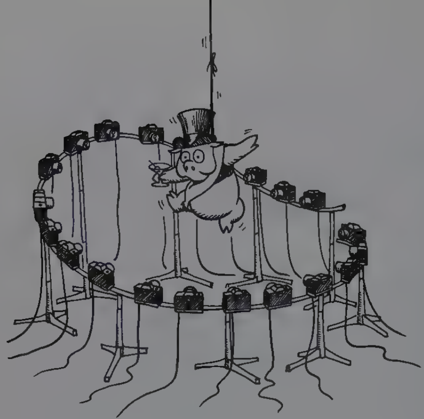
Crowds can be simulated with two- or three-dimensional techniques; the latter are described in Chapter 12. The crowds in both live-action movies ***Star Wars: Episode II*** and ***The Lord of the Rings: The***

Two Towers, for example, were created with computer-animated three-dimensional models (Fig. 12.6.1). Animated feature movies *ANTZ* and *The Prince of Egypt*, for example, also include three-dimensional crowds rendered in a non-photorealistic style (Figs. 2.7.11 and 12.6.2). The latter combines three-dimensional crowds with two-dimensional hand-drawn characters and hand-painted backgrounds. **Two-dimensional crowd replication** is a popular technique that starts with a few live-action elements—groups of actors or animals for example—and usually requires fewer production steps than its three-dimensional equivalent. The essence of two-dimensional crowd replication is to define a few "seed" elements of the crowd and then to duplicate them within the frame, making sure that their scale, motion, and depth matches the perspective projection of the overall scene. Figure 13.7.1 shows an implementation of the crowd replication technique that uses a few rows of extras as a point of departure for the effect, while Figure 13.4.2 uses small clusters of actors moving toward the camera.

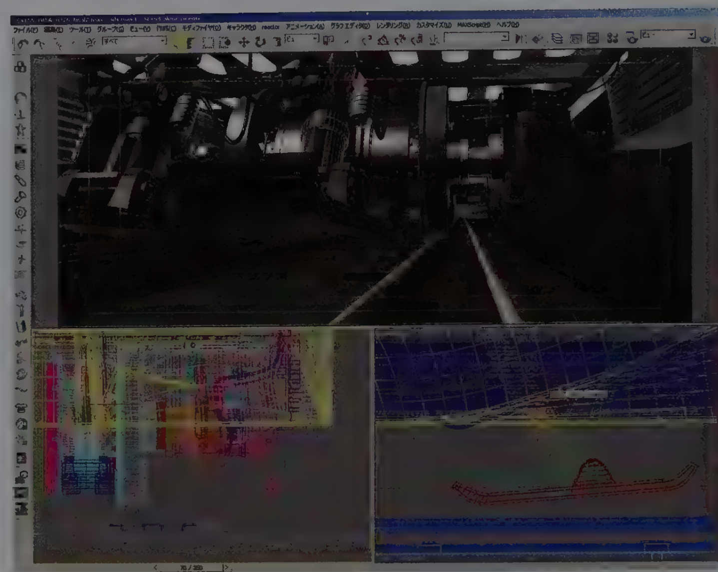
13.7.1 The army from the movie *The Emperor and the Assassin* was created with two-dimensional crowd replication techniques. The three source images above were included in the finished frame on the top of the page. (Images courtesy of Centro Digital Pictures Ltd.)

13.8 Three-Dimensional Morphing

Three-dimensional morphing is achieved by applying interpolation techniques to the geometry of three-dimensional models. Three-dimensional morphing techniques blend the shape of objects by interpolating the positions of their vertices in space. The rate of morphing can be controlled with functions such as time, speed, or proximity to a source. The most predictable three-dimensional morphing results are obtained when both objects being morphed have



13.11.1 Multiple still photography cameras are used to create the time freeze effect by recording the same instant from different points of view.



13.12.1 (Top right) Virtual train components modeled with 3D Studio and seen from a low camera angle, used in conjunction with the practical models shown in Figure 13.12.2. (Courtesy of Links DigiWorks Inc. © 2005 Fuji Television/ROBOT/TOHO/SPWT.)

the same number of vertices. (Fig. 11.2.6). Read Chapter 11 for more information on model and shape animation.

13.9 Motion Control

Before camera tracking techniques became feasible in the mid-1990s, using a motion control system was the only sure way to match a computer-generated animation with the movement of a live-action camera. A **motion control** system is a precise motorized crane that moves a camera along a specific path in response to a well-defined numerical input. A motion control system is also capable of repeating the same translations, rotations, and speeds over and over again. During the 1970s and 1980s, before computer animation and digital compositing became as prevalent as they are today, motion control systems offered the only solution for accurately repeating a specific move multiple times. Movies like *Star Wars* (1977) and *The Empire Strikes Back* (1980), for example, made extensive use of early motion control systems to film multiple scale models, each on a separate pass. Since the same motion path was followed each recording pass, the different elements were aligned and synchronized. Those early motion control systems were mostly used to shoot in **stop motion** mode, or a single frame at a time. Modern motion control systems can be used to shoot that way and also with continuous motion, without any loss of accuracy.

Motion control systems today can be used to feed the numerical information that describes their path to a computer animation system, but a three-dimensional path can also be laid out with a computer animation system and then fed to a motion control system for execution. This way the same motion path can be used to animate

13.12.2 The movie *Negotiator: Mashita Masayoshi* features a mysterious Tókyo underground train, and three versions of train were built. A full-scale train—being recorded as an element for a composite shot (top left), a miniature train, and a virtual train. (Courtesy of Links DigiWorks Inc. © 2005 Fuji Television/ROBOT/TOHO/SPWT.)

either the virtual camera or the real camera regardless of which one records first. An identical motion path will result in perfectly matching acceleration, speed, and direction of both the real and the virtual camera. The camera tracking techniques described earlier in the chapter are nowadays more commonly used for matching real and virtual motion than motion control systems.

13.10 Motion Capture and Virtual Characters

Motion capture is covered in Chapter 12, but it is mentioned again here because it is a technique commonly used in the production of visual effects and animation of virtual characters. In addition to the capture, editing, and blending of the captured motion, one of the key issues in achieving high-quality results has to do with the performance of the actor whose motion is being captured. In the early days of motion capture little attention was paid to this aspect, which is now recognized as the key driver of believable virtual characters. The character Gollum (Fig. 12.2.7) from the movie ***Lord of the Rings: The Two Towers*** was so vivid and had such powerful presence because the actor who "played" the part acted his motions taking into account the personality of the character and the limitation of motion capture. Pantomime techniques can be used to emphasize intention and to clearly delineate actions and movements. A virtual stunt double is shown in Figure 13.5.6.

13.11 Photogrammetry

Photogrammetry is a technique that can extract three-dimensional models from two or more still images of a subject. In its most gener-

13.12.3 (Next page) A list of the most influential visual effects films of all time, as compiled in 2007 by Hollywood's Visual Effects Society. Listings with an asterisk are tied for the same place.

al implementation, photogrammetry uses the still images of a subject to extract a depth map. That is converted to a polygonal model which can be textured with some of the original images used to create the model in the first place (Fig. 5.6.2). Image-based rendering, which is often used in conjunction with photogrammetry, is described in Chapter 5.

Time Freeze

The technique of **time freeze**, also known as time slice or **bullet-time**, is a variation of the photogrammetry technique. Time freeze achieves an effect of frozen time or very slow motion time by collecting multiple images of the same action from different points of view. More specifically, time freeze starts by placing an array of still cameras around or along a subject. ***The Matrix***, for example, used 120 cameras arranged in a circle and hidden behind a green screen wall with holes for the 120 lenses to see the action (Fig. 13.11.1). Triggering the cameras is computer-controlled so that it can be accurately timed. When all the cameras record the same instant from 100 different points of view, for example, each still frame can be edited in sequence and viewed in motion. In this scenario the still from camera 1 would become frame 1 in the sequence, camera 2 would be frame 2, and so on. When viewed in motion, this sequence of frames looks like a camera moving around a subject that is frozen in time. The cameras can also be triggered with a small delay, fractions of a second, between each one. When the still images are viewed in sequence the effect is of a moving camera around a subject frozen in time or one that moves in extreme slow motion, as the actor Keanu Reeves did when trying to avoid the bullets in ***The Matrix***.

13.12 Practical Effects

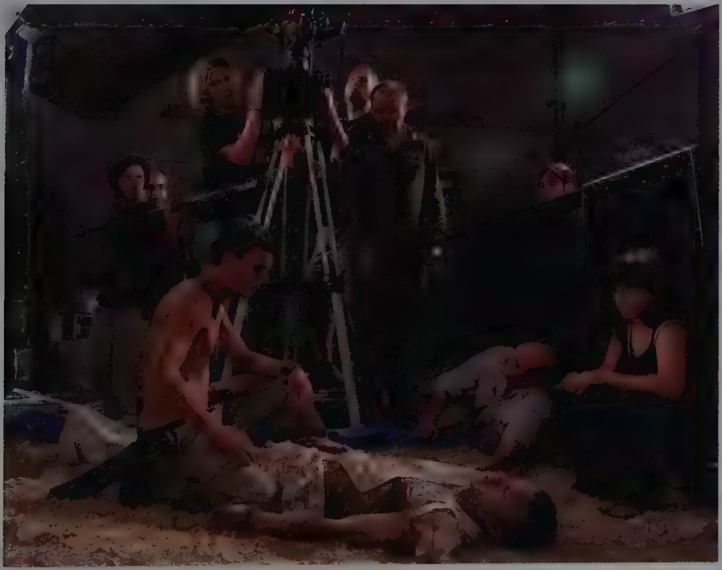
Practical effects are made in reality, not in the computer. Practical effects are created on locations, a sound stage, or a miniature set (Figs. 13.12.1 and 13.12.2). Practical effects, also called **special effects**, include real explosions (Fig. 14.2.1), stunts with or without wires (Fig. 1.5.4), models or miniatures, animatronics, facial and body make-up (Fig. 13.12.5), and prosthetics, among others. Today's practical effects are still an important component of the visual effects shot, and after being recorded on film or video they are usually enhanced further with digital tools, or combined with digital effects. Some of the most notable films with practical effects are included in the list issued in 2007 by Hollywood's Visual Effects Society. This list ranks the most influential visual effects films of all time, including films with practical and digital effects (Fig. 13.2.3).

The Most Influential Visual Effects Films

1. *Star Wars* (1977)
2. *Blade Runner* (1982)
3.* *2001: A Space Odyssey* (1968)
3.* *The Matrix* (1999)
5. *Jurassic Park* (1993)
6. *Tron* (1982)
7. *King Kong* (1933)
8. *Close Encounters of the Third Kind* (1977)
9. *Alien* (1979)
10. *The Abyss* (1989)
11. *The Empire Strikes Back* (1980)
12. *Metropolis* (1927)
13. *A Trip to the Moon (1902)*
14. *Terminator 2: Judgment Day* (1991)
15. *The Wizard of Oz* (1939)
16. *Who Framed Roger Rabbit* (1988)
17. *Raiders of the Lost Ark* (1981)
18. *Titanic* (1997)
19. *Lord of the Rings: The Fellowship of the Ring* (2001)
20* *Jason and the Argonauts* (1963)
20* *E.T. the Extraterrestrial* (1982)
22. *Toy Story* (1995)
23. *Pirates of the Caribbean: Dead Man's Chest* (2006)
24. *The Ten Commandments* (1956)
25* *The War of the Worlds* (1953)
25* *Forrest Gump* (1994)
25* *Citizen Kane* (1941)
25* *The Seventh Voyage of Sinbad* ('58)
25* *20,000 Leagues Under the Sea* ('54)
30. *The Terminator* (1984)
31. *Aliens* (1986)
32. *Mary Poppins* (1964)
33. *Lord of the Rings: The Return of the King* (2003)
34. *Forbidden Planet* (1956)
35. *Babe* (1995)
36* *The Day the Earth Stood Still* (1951)
36* *Lord of the Rings: The Two Towers* (2002)
38. *King Kong* (2005)
39. *Planet of the Apes* (1968)
40. *Fantastic Voyage* (1966)
41* *Jaws* (1975)
41* *Ghostbusters* (1984)
43. *Sin City* (2005)
44. *Superman: The Movie* (1978)
45. *Snow White and the Seven Dwarfs* (1937)
46* *The Lost World* (1925)
46* *Return of the Jedi* (1983)
48. *What Dreams May Come* (1998)
49. *American Werewolf in London* ('81)
50* *Darby O'Gill and the Little People* (1958)
50* *The Fifth Element* (1997)

13.12.4 A visual effects shot with live actors composited into computer-generated kangaroos. (Image courtesy of Andy Boyd, www.a3d.co.uk.)

(Next page, left: An actor filmed on blue screen is composited with layers of practical fire, particle systems, a live stadium, and a few distortion filters to simulate the heat and force wave running through his arms, face, and torso. Images courtesy of Centro Digital Pictures Ltd.)



13.12.5 (Top left) Detail of a still frame from *Linko* shot on green screen with a Red One digital camera at 4K resolution, and composited onto virtual sets. (Top right) A tracking shot on a green-screen soundstage. (Above) A visual effects supervisor measuring incident light on an actor. (Left) Recording a performance on the set. (© IB Cinema. All rights reserved.)

CHAPTER 13

Key Terms

2K file
Background plates
Background replacement
Blue screen
Bullet time
Camera reports
Camera tracking
Camera stabilization
Character extensions
Chroma key
Cinematographer
Clean background plate
Color spill
Computer-generated props
Compositing supervisor
Computer animation supervisor
Cross-over techniques
Digital still camera
Director
DP, DOP
Film scanning
Generalists
Green screen
Lighting markers
Match move
Match reality
Megapixels
Metadata
Motion control
Motion path
Practical effects
Previz
Production pipeline
Rotoscoping
Salad of effects
Second unit of photography
Set extensions
Sequence supervisor
Shot breakdown
Special effects
Specialists
Survey reports
Temps
Three-dimensional rotoscoping
Time freeze
Tracking markers
Two-dimensional crowd replication
VFX
VFX elements
Visual effects core team
Visual effects producer
Visual effects supervisor
Visual look
Wire removal

(A skeleton and muscle simulation driven by Absolute Character software. Courtesy of Snoswell Design.)

SECTION V

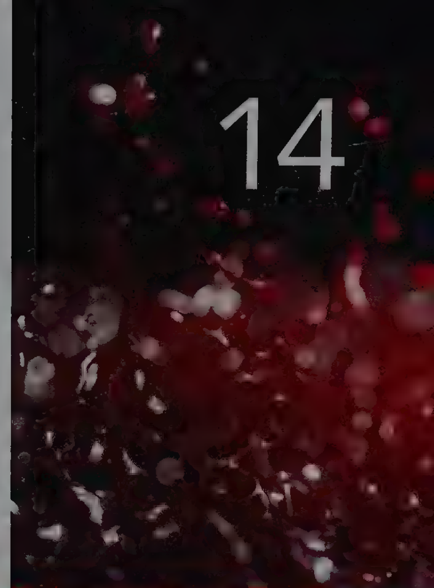
Compositing and Output

(Previous page) This bird's-eye view of an architectural project is a composite of computer-generated visualizations of buildings with a photograph of the actual location. (90 High Holborn. © Hayes Davidson.)

14.1.1 For this shot where the character surprises us with unexpected poses, the computer animation was composited into live action backgrounds. The camera point of view accentuates the comedy of the moment. (© 2002 Blockbuster Entertainment/Tippett Studio.)

CHAPTER 14

Retouching, Compositing, and Color Grading



Summary

ONCE THE THREE-DIMENSIONAL SCENES have been rendered, the resulting two-dimensional images can be enhanced with further digital processing. Image retouching and compositing are two of the most popular postprocessing techniques, along with color balance and correction and image sequencing.

QUICK INDEX

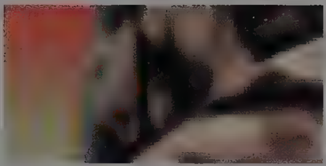
14.1 Basic Concepts of Image Manipulation

There are myriad techniques for enhancing and combining different live-action elements, two-dimensional renderings of three-dimensional environments, and combinations of live and computer-generated imagery. **Postprocessing** has its roots in traditional retouching just as three-dimensional computer modeling can be compared to the work of a sculptor working with traditional materials, and the rendering process can be compared to the work of lighting designers, makeup artists, photographers, and painters. Postprocessing can be compared to the work that photographers do in the darkroom, once the images have been recorded on film but before the final prints are made. **Image manipulation** techniques are used to modify the color, contrast, and brightness of images, as well as their content. These techniques can increase the overall brightness of an image or lower the contrast between light and dark tones. Techniques like retouching and compositing can also be used to remove mistakes or to combine areas of different sources into a single image (Figs. 14.1.1 and 14.1.4). Some of the basic concepts of image manipulation techniques include pixel and color resampling, editing with parameter curves and histograms, and compositing with alpha channels and transition effects.

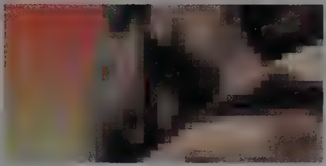
Resampling the Pixel Resolution

One of the key techniques of image manipulation consists of changing the spatial resolution of an image. This is called **resampling an**

(Top: Detail of *Shatter*. Image courtesy of Kouhei Nakama.)

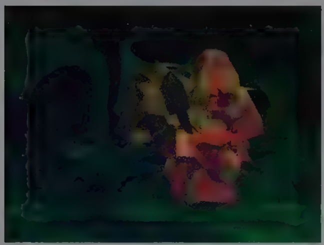


WEIGHTED PIXEL INTERPOLATION

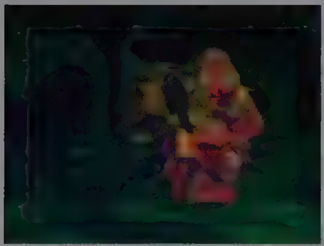


LINEAR INTERPOLATION

14.1.2 Details of an image originally rendered at 72 ppi and resampled to 300 ppi with weighted pixel interpolation (top) and with linear interpolation (bottom). The transitions and blending of resampled pixel color values are smoother when weighted pixel interpolation is used.



269,400 PIXELS AT 300 PPI



15,552 PIXELS AT 72 PPI

14.1.3 More pixels are required for higher resolutions. A 72 ppi render requires only 15,552 pixels, while a 300 ppi render at the same size requires 269,400 pixels.

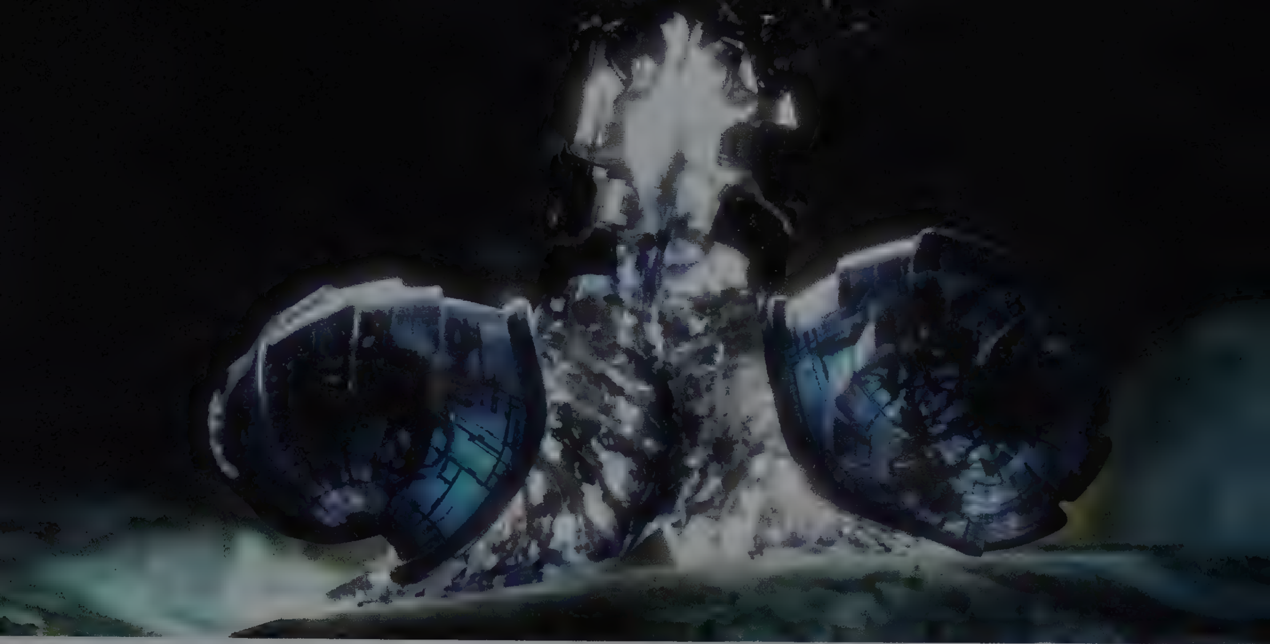
image and is often used when the dimensions or the spatial resolution of an image need to be increased or decreased. It is necessary to resample an image, for example, when a scene has been rendered at a resolution of 72 **pixels per inch** (ppi) but the final output requires 300 dots per inch (dpi), or when a scene was rendered at a size of 1,000 × 1,000 pixels, but the client needs it at 500 × 500 pixels.

The continuous values of a live scene or a photograph are sampled when the live scene is digitized with a digitizing camera, or when a continuous tone photograph is scanned with a scanner. In both cases, the frequency of the sampling determines the resolution of the digital image. Many samples, or point measurements, of color result in many pixels. A few samples yield few pixels and a low resolution. When images are resampled, the information contained in a digital file can change subtly or dramatically. When images are sampled down, the software averages the values of several pixels and discards some of the original information. When images are sampled up, the software averages the values of several pixels and creates new information. In either case, this averaging of pixel values is based on one of many **interpolation techniques** that create new pixel values by averaging the existing pixel values in different ways.

Simple interpolations, such as **neighbor pixel interpolation**, determine the value of a new pixel by averaging the values of only two pixels. When an image is resampled to double its original size, for example, the number of pixels that has to be created is twice the number of original pixels. With a simple interpolation technique each new pixel is created between two existing pixels, and its value is determined by averaging the value of just those two pixels. Resampling based on simple or linear interpolation techniques usually creates images with small defects like banding, aliased edges, or loss of detail.

There are more sophisticated resampling techniques on the other end of the interpolation spectrum. Some determine the value of a new pixel by averaging the values of many pixels and assigning each of them a priority or weight based on their proximity to the new pixel being calculated. An example of a **weighted interpolation** technique is illustrated in Figure 14.1.2 where the values of several pixels are averaged to create a new single pixel. When an image is sampled down, some of its original information is lost, and the resampled image will look quite different from the original if it is resampled up again after having been resampled down.

Regardless of the pixel interpolation method used, when resampling an image it is important to determine whether maintaining the **file size** (the amount of space it takes to store it) is important. If keeping the file size constant is an issue, the physical **image dimensions** (height and width) of the file will surely change. Inversely, if the file size may change as a result of the resampling, then the dimensions remain constant. For example, a 46 KB image that measures 2 × 1.5 in. at 72 ppi can be resampled at 300 ppi and keep its file size of 46 KB constant but the dimensions change to .48 × .36 in. If the file size must remain constant, the absolute number of pix-



els also remains constant, but the image dimensions change because at the higher resolution more pixels are required to create an inch. If the file size can grow, so does the resolution and the number of pixels, while the dimensions remain unchanged. But if the 72 ppi image would be resampled or rerendered to the dimension of 2×1.5 in. at the increased resolution of 300 ppi, its file size would increase to 792 KB (Fig. 14.1.3).

14.1.4 The light glow effect on the snail characters and the depth of field in *458nm* was added to the render with Fusion compositing software. (© 2006 Filmakademie Baden-Württemberg, Jan Bitzer, Ilija Brunck, and Tom Weber.)

Resampling the Color Resolution

Another useful technique for manipulating images of three-dimensional rendered scenes consists of changing, or resampling, their color resolution. Satisfactory results when **resampling the color resolution** of a rendered image are obtained only when it is necessary to lower the color resolution but not when it is necessary to increase it. Increasing the color resolution of an image can be achieved with good results only by rerendering the three-dimensional scene at a higher color resolution. Color resampling is useful—and even necessary—when the image has been rendered at high color resolutions but needs to be displayed at a low resolution. This would be the case, for example, of renderings created at a resolution of 32-bit color for the purpose of feature film presentations that are adapted to the home videogame format and had to be resampled down to 8-bit or 16-bit color to be displayed efficiently. Resampling the color resolution of entire projects is common now that many works and intellectual properties previously developed for high-resolution formats are being released in home entertainment systems at lower color resolutions.

Most three-dimensional computer rendering systems create 24-bit or 32-bit color images. When color resampling is needed the

14.1.5 An image at four different color resolutions. The full-color version of the image, at 8-bits per channel, contains millions of colors. A version with only 256 colors is created with a color look-up table at 8 bits per pixel. A 4-level color look-up table, at 4 bits per pixel, displays one of 16 shades of color. Dithering techniques can improve the color accuracy of a limited color palette. The image spatial resolution is 2300 × 2300 pixels. See Figs. 15.2.3 and 15.2.4 for additional information on color look-up tables (Photo by Jennifer Beecher.)

FULL COLOR, 8 BITS PER CHANNEL

256 COLORS, 8 BITS PER PIXEL

16 COLORS WITH DITHERING

16 COLORS

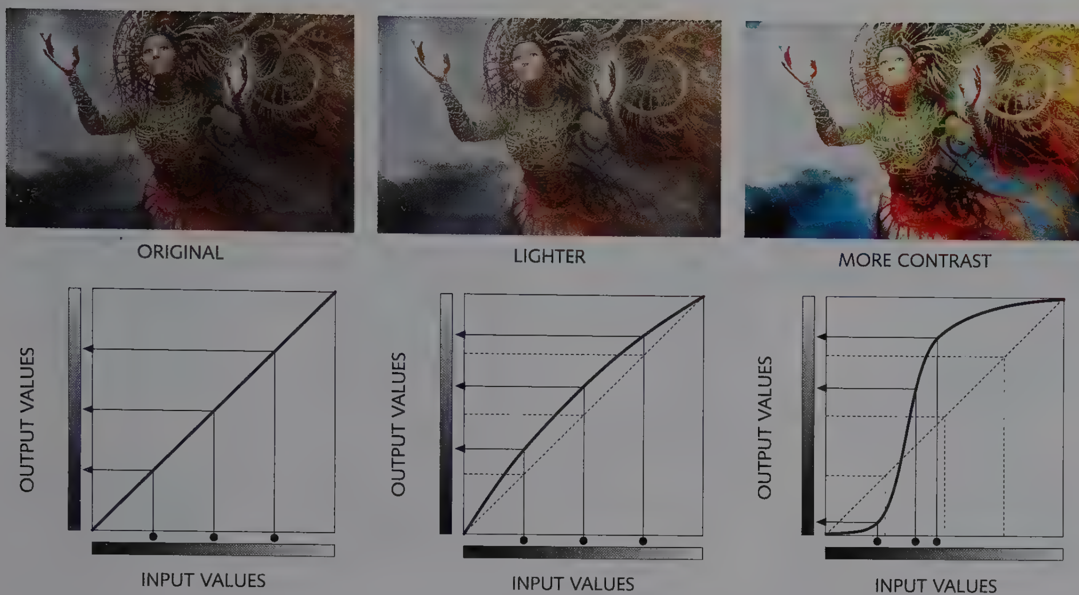
FULL COLOR / INDEXED COLOR

14.1.6 A full color image (left half) that is converted to indexed color (right half) can retain a fair amount of color fidelity and crispness with a custom color look-up table. Color detail was retained in this conversion, except for a slight increase in contrast and minimal compression or flattening of the darkest tones in some hues. (Screen shot from *Myst* CD-ROM computer game. © 1993 Cyan, Inc. All rights reserved.)

process is very straightforward and is usually implemented in the form of choices listed in a dialog box or a pull-down menu. The most common resolutions for color images include 32-, 24-, 16-, and 8-bit color (Fig. 14.1.5). But the loss in color detail due to color resampling can be severe—for example, when going from the 16 million colors of 24-bit color to the 256 colors available at 8-bits per pixel. For this reason, the techniques of color dithering and color look-up tables are often used to minimize the artifacts and loss of detail created by color resampling. Both of these techniques are especially effective when 8-bit color is utilized.

Color dithering simulates shades of color with dot patterns of several colors. Dithering preserves some of the color detail that tends to be lost when full color images are sampled down to 4-bit per pixel color. But at the same time, dithering lowers the apparent spatial resolution of an image because the dot patterns used to simulate shades of color often look like enlarged pixels (Fig. 14.1.5).

Color look-up tables, also called **indexed color**, are a popular technique that is a carry-over from the early days of interactive two-dimensional videogames. A **color look-up table** is a limited palette of colors that represents quite faithfully a much larger selection of colors. Color look-up tables contain a tight selection of colors (256 in the 8-bit color mode) that gives the impression of a much larger palette. The main purpose of color look-up tables is to optimize



14.1.7 The unretouched original image displays its brightness function curve as a straight diagonal line (top left). When this curve is moved up the output values will be higher than the input values. If the histogram controls brightness, for example, the output values will be brighter than the input values. The brightness of the image increases in the midtones when the function curve is pulled up (top middle), the smooth curve results in a gradual change of the brightness values. Abrupt changes in the curve result in abrupt changes in the brightness values (top right)—in this case, flattening the lightest and darkest values and compressing the midtones. (Image courtesy of Meats Meier.)

color accuracy in a low color resolution environment. When a full-color RGB image or sequence is converted to indexed color, there are different methods for choosing the colors that go in the look-up table. These methods include using a generic color look-up table or customizing the color manually or automatically.

It is convenient to use a generic color look-up table, often called a **system palette**, because it is a standard feature of most image manipulation software programs. A generic color look-up table produces acceptable results with images that have a balanced distribution of color. But generic palettes usually produce poor results when the colors in the sequence are biased toward a single hue—for example, a scene at dusk with mostly dark colors. In such cases, generic palettes usually lack the variety of color that is required to represent the subtle variations of color within a narrow chromatic range. One of the advantages of using a generic color look-up table is that it reduces memory requirements and increases performance because the generic color look-up table can be used with a variety of different images. In an indexed color environment, a generic look-up table only has to be loaded once for all images, as opposed to customized palettes which have to be loaded every time the image is used.

Custom color look-up tables can be built for a specific image or sequence by *manually* selecting the colors that best convey the variety of colors contained in the original full color version. Custom color look-up tables are best when they contain the most commonly used colors throughout the sequence for which they are built (Fig. 14.1.6). Custom color look-up tables can also be built *automatically* by letting the software decide which limited color palette would best represent the thousands of subtle colors contained in a scene.

14.1.8 This histogram represents the pixel distribution throughout the tonal range of the original image . The continuity of the vertical lines in the graph reflects the continuity of levels in the original image.

14.1.9 (Opposite page) Two original images were composited using eight different masks in the alpha channel. The background is a photograph of the tropical rain forest, and the foreground is a synthetic rock. In this masking setup the black areas in the mask block or cover the green background, and the white areas let it show through.

Parameter Curves

The parameter or function curves are graphs that represent and control different attributes of an image, such as brightness or color. These attributes can be easily modified by manipulating the function curves without having to alter the image directly with a retouching tool. Making image manipulations that involve all of the image or large portions of it are best performed with function curves. The parameter curves used for image manipulation are similar to those used for controlling animation interpolations (read Chapter 11 for more information on animation interpolation).

Parameter curves for image manipulation are usually represented by a line that starts at the lower-left corner of a square and ends at the upper right corner. The **straight diagonal line** (in a parameter curve) usually represents a linear relation between the input and the output values (Fig. 14.1.7). Any changes made to the linear parameter curve will result in changes to the image. Generally, if the line is pulled above the straight diagonal path the attribute increases, and if the line is moved below the diagonal path the attribute being controlled by the line decreases. The manner in which the parameter curve is redrawn also influences the way in which the attribute is controlled. If the parameter curve is redrawn with a series of **soft curves**, the attributes of the image change gradually as in a curved interpolation. If the parameter curve is redrawn with a series of **angular straight lines**, the attributes change abruptly as they sometimes do at the junction of two linear interpolations.

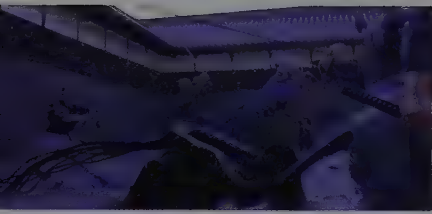
Linear Color Space and Nonlinear Color Space

In simple terms, the concepts of linear and nonlinear color space are often used when it is necessary to convert images to a smaller file size while preserving as much image resolution as possible. A **linear color space** is used for direct and simple conversion—for example, a mapping of 16 levels of gray that yields 8 levels that are numerically equally spaced. A **nonlinear color space**, also called **log space**, is used for conversions that seek to preserve certain areas or levels of the image, for example a mapping of 16 levels of gray that yields 8 levels that are perceptually equally spaced. Nonlinear color spaces are commonly used in visual effects production to store subtle information scanned from film negative or interpositive.

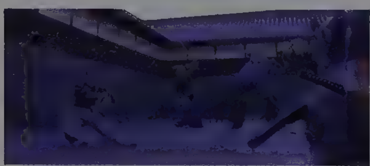
The Histogram

One of the principles of traditional painting and classic photography recommends the creation of images with a rich tonal range. In a monochromatic image, that means using a sample of gray values that is evenly and continuously distributed across the grayscale. In a color image, a rich tonal range includes color values as well as brightness and contrast. An image, a black and white photograph for


BACKGROUND
FOREGROUND
ALPHA CHANNEL WITH MASK
COMPOSITE

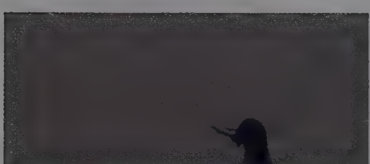


FINAL COMPOSITED FRAME

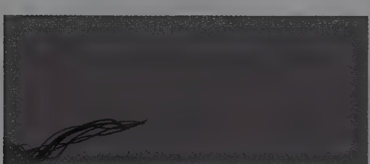


RETOUCHED LIVE ACTION LAYER

BODY AND CLOTH LAYER



CHARACTER HAIR LAYER



CHARACTER EXTENSIONS LAYER

14.1.10 Frame from *Shinobi* showing the layers, added to the live-action frame, containing the character's body, cloth, hair, and extensions. See Figure 2.7.6 for the previsualization and additional frames from this sequence. (Courtesy of Links DigiWorks Inc. © 2005 SHINOBI Film Partners.)

example, that contains small amounts of pure black and pure white plus a continuous and even distribution of gray values usually has a wider expressive range than a mannerist image that dwells exclusively on the dark or light tones of the scale.

Looking at the histogram of an image is an effective way to analyze the distribution of values in each of the channels of the image. A **histogram** consists of a graphical representation of the distribution throughout the grayscale of the pixels that constitute an image (Figs. 14.1.8 and 14.2.2). Histograms can be used to adjust both the input and the output level values of an image. In an 8-bit monochromatic image, the horizontal line in a histogram represents the range of gray values between pure black, usually represented with a value of 0 on the left side, pure white, represented with a value of 255, and the midtone represented with a 1. In a full-color image, RGB for example, each component color has a histogram. The combined component colors can also be controlled with a single histogram. The number of pixels that have intensity values between black and white are represented with a vertical line or bar. The larger the number of pixels on a single value, the taller the line. These lines are located throughout the horizontal line in accordance with the pixel values contained in the image. An image with pixels that are evenly distributed throughout the grayscale has vertical lines all across the histogram, while an image with only dark and light values—but no midtones, for example—has vertical lines on the extremes of the histogram but not in its center.

Image Layers

Working with multiple **image layers** or **channels** is similar to the process of painting on transparent acetate overlays that is still used in traditional cel animation. In this case, different components of the image are painted each on a cel, and the finished image can only be seen when all the layers are assembled in sequence (Fig. 14.1.10). But when each layer or cel is viewed separately, only a part of the image is visible. The principle of using image layers is widely used in multi-pass rendering and is also the basis of four-color separation and a staple process in the graphic arts (Fig. 6.2.1).

Image layers are used for a variety of image manipulation purposes. As previously mentioned, image layers are also used to display separately each of the component or primary colors of an image in several color modes including RGB, CMYK, and HSL. It is possible to work on a single layer separately from the other layers—for example, increase the brightness or apply a filter to a single layer and then merge the layers again and view the combined results.

The Alpha Channel, Masks, and Mattes

Image layers can also be used to composite and blend the contents of several layers and place the resulting image in a separate set of

image layers. This process is called image compositing or blending, depending on the specific techniques used to combine the contents of the image layers. Image compositing is also known as **matting** in the film industry, and **masking** in the graphic arts industry. One or several **alpha channels** are commonly used to aid in the compositing of multiple layers. Alpha channels typically contain a black-and-white image, called a **mask**, that masks or protects select parts of one or several of the layers being composited. The mask, or **matte**, contained in the alpha channel is like the **stencils** traditionally used to label wooden cargo containers or to aid in the creation of images with the delicate techniques of airbrushing or silkscreen. In either case, the stencil is perforated with a design or with letterforms and placed on a surface. The solid areas of the stencil protect the surface underneath and the perforated areas allow the paint to reach the wooden plank in the case of the cargo container, or the paper surface being airbrushed or silkscreened. Compositing with a mask in the alpha channel allows select portions of images to be composited as **foreground elements** in front of a **background**. Figure 14.1.9 illustrates the compositing of a foreground image with a background, using all the possible combinations of mattes derived from the simple foreground shapes and the background.

14.2.1 These frames show (from top to bottom) the original shot with the full bridge, the retouched bridge to make it seem more dangerous, the composite with an explosion, and the retouched bridge again after the explosion. (*Le Bossu*, a film by Philippe de Broca. Production: ALICÉLÉO. Visual Effects: Ex Machina Paris. Images courtesy of Ex Machina.)

14.2 Image Retouching

Digital image manipulation techniques can provide the best that both the **darkroom** of a photographer and the **studio** of a painter have to offer. For example, with image manipulation software, we can fine-tune the tonal range of rendered images, adjust their contrast and brightness, and apply digital filters to create special effects.

Postprocessing programs facilitate the blending of computer-generated images with photographic or painted images. The final result can be output onto a variety of media only after the two types of images are combined with integrated tools and techniques. These capabilities have opened new creative avenues and experimentation that was not possible with traditional tools.

Before computers came along, it was impractical to combine painting with photography because each occurred in very different, almost incompatible, media. Bringing the photographic image onto the canvas was cumbersome, and painting on photographic paper was limiting. The narrow focus of the traditional tools did not help either. Brushes and paint did not work well on photographic images, and many of photography's essential tools such as lenses and filters could not be used to paint. The closest these two media ever got before computers was retouching photographs for commercial advertisements or artistic collages. In the former case, the process of blending the painted image with the photograph was painstaking, limited, and extremely expensive. In the latter case, the painted image and the photographic image usually remained quite independent from each other, and realistic representation was rarely a priority.

14.2.2 A histogram editor can be used to redistribute the color values in an image. The shadow editor can be used to increase the dark values in the image. The midtone editor can also be used to increase the dark values in a different tonal range. Both the highlight and midtone editor are used in this example to increase the overall highlight values in the image. (Image courtesy of Meats Meier.)

Retouching Tools

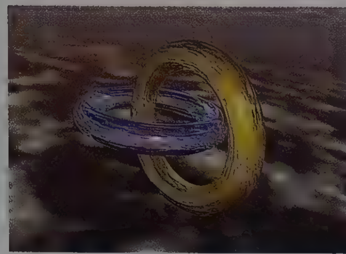
A wide variety of tools is available for **retouching** rendered and live-action images. Some of these digital tools are inspired by the tools of painters and illustrators, and others are based on the tools used by photographers. Retouching tools are typically used to touch up small mistakes in the rendered file or to add details that are missing (Fig. 14.2.1). As a general rule, when mistakes are made in the modeling or rendering process, it is best to model and render the scene again. But in a few cases, the production schedule or budget does not permit fixing of minor mistakes by remodeling or rendering again. This is especially true when the time it would take to render is beyond the production scope of the project or in terminal situations when, for example, the computer-generated animation has already been composited with live-action material and the director changed his mind about a certain coloring or lighting in the scene.

Some of the digital retouching tools that are inspired by **traditional painting tools** include brushes, pencils, and rubber stamps. These painting tools are used to paint over selected areas of the rendered image, and their simulated attributes, such as width or pressure, can be customized. In addition, the way in which these paint tools deposit the virtual paint on the digital image can also be customized in many different ways. The digital paint, for example, can have different degrees of transparency or can affect only a certain range of pixels with a certain color or brightness value. A variety of simulated paper textures can also be applied to the surface of the rendered image.

The retouching tools based on **photographic procedures** include a tool for lightening or darkening selective parts of the image interactively, just as dodging and burning are performed on photographs as the image on the film negative is projected onto

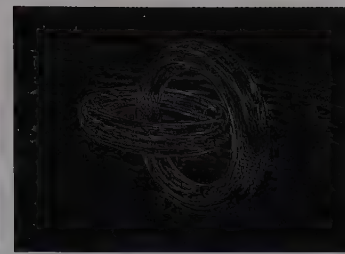
0	0	0	0	0
0	0	-1	0	0
0	-1	5	-1	0
0	0	-1	0	0
0	0	0	0	0

SHARPEN



0	0	0	0	-1
0	0	0	-1	0
0	0	5	0	0
0	-1	0	0	0
-1	0	0	0	0

EMBOSS



-1	0	0	0	-1
0	0	0	-1	0
0	0	5	0	0
0	-1	0	0	0
-1	0	0	0	0

EDGE DETECTION

14.2.3 In this example of custom filters applied to the top image in Figure 14.2.6, the cel in the center of the 5 × 5 matrix contains the value that is to be used in the mathematical operation applied to the brightness of the pixel being evaluated. The cels adjacent to the center cel represent the pixels immediately surrounding the pixel being evaluated. The outer cels represent a second group of pixels that are a pixel away from the one being evaluated. The values used in each of the filtering variations are shown below each image. The results include sharpening, embossing, and edge detection.

photosensitive paper. The histogram editor itself and all the parameter curves are also retouching tools that are rooted in the photographic tradition because they help define an image by adjusting the tonal and contrast settings of the image.

The **digital retouching tools** that are truly unique, even exclusive, to computers are those for selecting different parts of the image. These tools provide techniques for selecting the pixels or areas of the rendered image that need to be retouched. The most common selection tools provided by retouching software are those that let the user enclose the selected area with a tool like a marquee or a free-form lasso. The more sophisticated selection tools allow the selection of image areas based on their pixel values. This method of selection works on contiguous pixels as well as pixels scattered throughout the image. All the pixels in an image within a specific color range can be easily manipulated or replaced with this selection method.

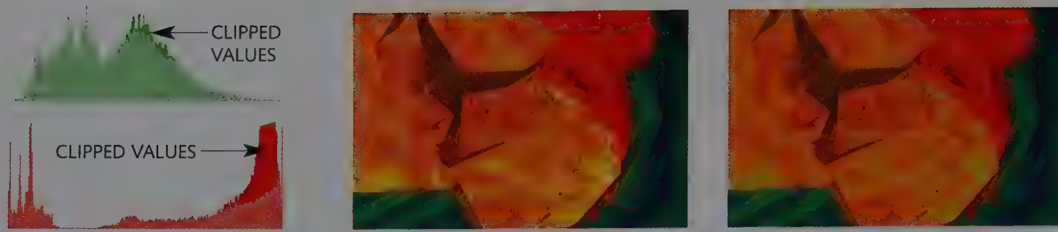
Editing the Tonal Range

One of the most useful tools provided by many image manipulation programs is the histogram. The histogram is a graph that shows the distribution of light, middle, and dark values in an image. Using a **histogram editor** makes it possible to modify with extreme accuracy the tonal range of an image, which defines many of the characteristics that give images their distinctive character. **Tonal range** includes the distribution of values throughout the grayscale and the relation between the highlights, midtones, and the shadows, the brightness and contrast levels, and the color balance of images.

The histogram in Figure 14.1.8 shows an uneven distribution of pixel values across the tonal range represented by the width of the horizontal line. Each of the vertical lines represents the number of

BLUR

14.2.4 A blurring filter applied to the source image, top in Figure 14.2.6.



CLIPPED RED AND GREEN COLORS

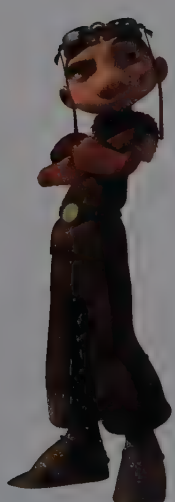
BEFORE NTSC FILTER

AFTER NTSC FILTER

14.2.5 Before and after an NTSC filter was applied to the image to remove the RGB colors (particularly the bright yellows) that are beyond the chromatic range of NTSC video. The dark colors in the histograms show the red and green hues that were clipped by the filter.

pixels in the image at each of the tonal values between black and white, and the higher lines represent the most number of pixels at that tone of gray or particular color. The three triangles below the tonal range represent (from left to right) the markers of shadow, midtone, and highlight values. Their normal settings are 0, 1, and 255, respectively. These markers are used to redistribute the pixel values across the tonal range. By sliding any or all of the markers or editors to the right, for example, the darker values are given more prominence in the tonal range. Moving just the **shadow** editor to the right relocates the position of the pure black to a higher position in the tonal range (from 0 to 31), therefore increasing the proportion of dark values in the image (Fig. 14.2.2a). Moving the **midtone** editor to the right expands the presence of the dark values within the tonal range (Fig. 14.2.2b). Moving both the **highlight** editor and the midtone editor to the left, for example, increases the range of the light values in the image (Fig. 14.2.2c).

Using a histogram to effectively edit the tonal range of an image requires practice. A more straightforward method for editing the tonal range of an image is based on using simpler controls for **brightness** and **contrast**. These controls are often in the form of sliders, parametric curves, or fields that accept numerical values typed directly on the keyboard.



(Character from *Dragon Hunters*. © MMVII Futurikon Films, Trixter, LuxAnimation, France3 Cinéma, RTL-Tvi, in coproduction with Mac Guff Ligne.)

Digital Filters

Digital two-dimensional filters are rarely applied during the actual three-dimensional rendering process. These filters are usually applied to the two-dimensional images of three-dimensional scenes after the scenes have been rendered. Like their photographic relatives, digital filters modify the appearance of an image, but are able to change many more attributes and with much more precision than it is possible to change with photographic filters. Digital filters can be applied selectively to all of the image or only some areas. This capability turns digital filters into extraordinary retouching tools. Digital filters cover a wide range of special effects ranging from a simple blurring filter to a compound filter that adds lens flare and motion blur.

Digital filters work by submitting single or groups of pixels in the image to a series of mathematical operations. Each type of filter is based on a unique combination of mathematical operations that

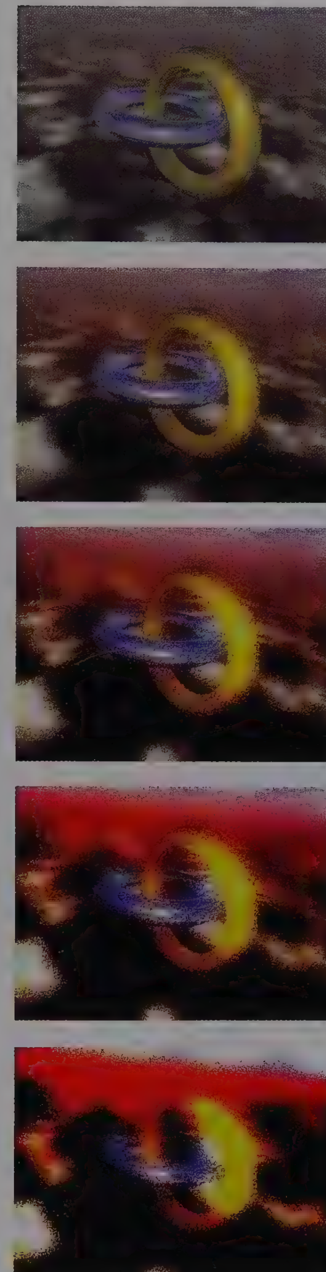
process the numerical values of pixels, both independently and in relation to the neighboring pixels. The matrix and values of a **custom filter** that convolutes, or twists, the brightness of the pixels in the image are shown in Figure 14.2.3.

Sharpening filters are commonly used to increase the contrast of adjacent pixels in areas of an image that may be blurred, for example, due to the lack of lighting during the rendering process. Sharpening filters can also be used to bring out more detail in surfaces that have been texture mapped or in images that have been resampled to a higher resolution. These filters use a variety of sharpening techniques that increase contrast based, for example, on differences of color or brightness, or only where edges of shapes are found. Even though sharpening filters are applied to every pixel of the selected image area, in some cases a **filter radius** is specified to determine the size of the filter as it looks for edges and determines how much sharpening should be applied (Fig. 14.2.3).

Blurring filters can be used to soften the areas of rendered images where too much contrast between adjacent pixels creates jagged edges or **texture noise**. Blurring usually works by bringing the intensity or color values of adjacent pixels closer to one another (Fig. 14.2.4). On occasion, blurring filters are used as an antialiasing retouching tool. Blurring filters can also soften the sharp edges of polygonal renderings, and the edges of masks. This technique is very effective for eliminating the jagged edges and color aliasing that happen during compositing when the mask is too sharp and displays jagged edges.

A few additional filters that can be quite functional include edge detection filters and NTSC (National Television Standards Commission). **Edge detection filters** identify the edges of a shape, isolate them, and even trace a contour around them. When combined with selection tools, both of these techniques are useful for creating the masks used in the compositing of images. Edge detection filters create tight masks quickly and avoid the repetitive manual work that is otherwise required to create masks (Fig. 14.2.3). An **NTSC color filter** clips from the image the RGB colors that are beyond the chromatic range of the NTSC video signal (Fig. 6.2.5). NTSC color filters are useful in maintaining the quality of color, and in making sure that colors are not too hot—beyond normal limits—for the video standard (Fig. 14.2.5). (Read Chapter 15 for more details on NTSC.)

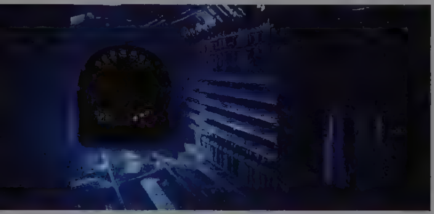
In addition to the basic digital filters, dozens of striking visual effects can be achieved by filtering an image. A select group of these filters helps to fine-tune and refine renderings of three-dimensional scenes. But the majority of filters have such an overwhelming effect on the image that they are better suited to aid in the creation of special effects that visibly alter the original renderings or to prepare the two-dimensional images that are used as image maps. When used appropriately, even the filters that distort the image can create startling effects like bold changes in color, sensuous undulations and ripplings, delicate embossing and contouring, tingling textures, and faceted brushstrokes (Fig. 14.2.6).



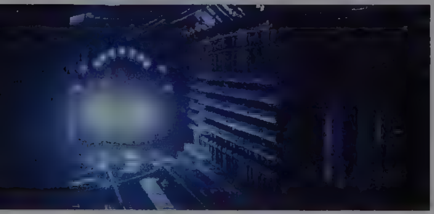
14.2.6 This sequence of images was created by applying the same filter to the initial image and to each of the subsequent images.

SHOT OF REAL TRAIN

REMOVE REAL REFLECTORS



ADD SECONDARY LIGHTS



FINAL COMPOSITE

14.2.7 The brightness and density of the lights in the real underground train was improved in compositing by adding secondary practical lights and boosting the highlights. (*Negotiator: Mashita Masayoshi. (*Courtesy of Links DigiWorks Inc. © 2005 Fuji Television/ ROBOT/TOHO/SPWT.)

14.3 Image Compositing and Blending

Image compositing consists of combining two or more different images into one in such a way that an illusion of time and space is created: the illusion that all the elements in the image were recorded together at the same time and place (Figs. 13.2.4 and 14.2.7). Image compositing created with traditional tools—such as scissors, glue, and paper—is called **collage**, an assembly or composition of image fragments or materials from different sources. The word collage has roots in the French word *coller*, which means "to glue."

One of the main purposes of image compositing is usually to save expensive production costs or to simulate something that is physically impossible to create in our reality—for example, a family and their pets having a picnic on the surface of Saturn while their chauffeur drives their spaceship through the rings around the planet.

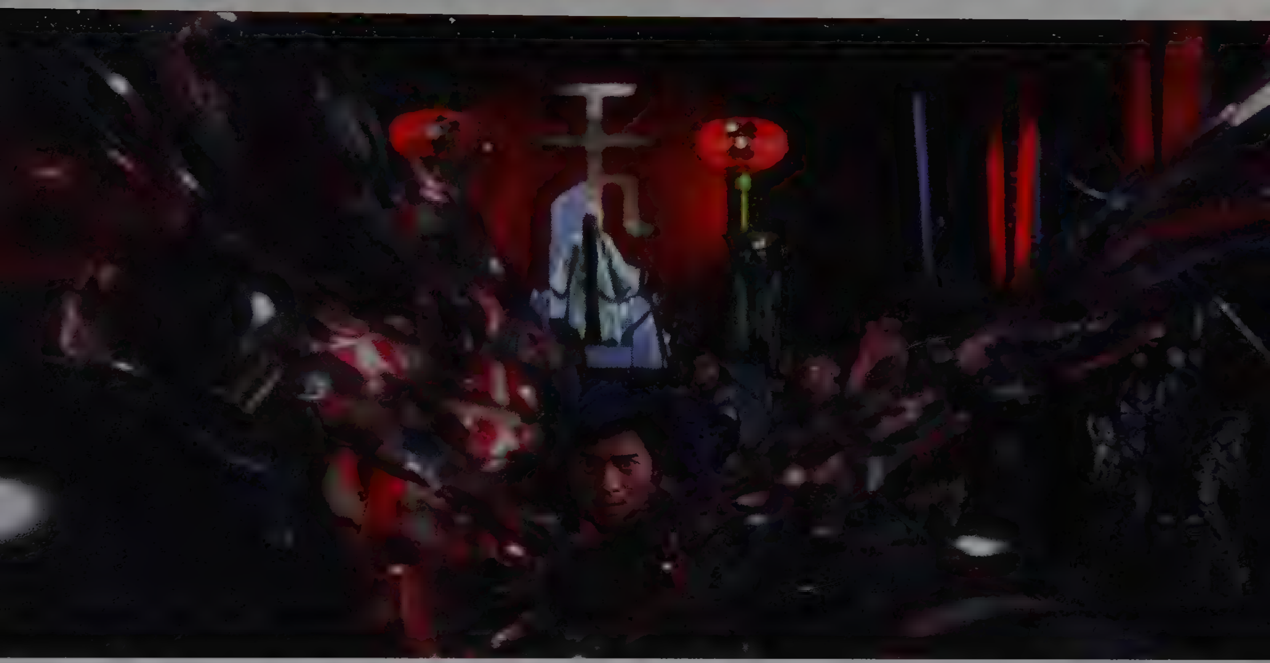
The process of compositing images from different sources into a single visually coherent image can be performed on both still and moving images. Still composites are often called collages, while moving composites result in dynamic composites or transition effects. Combining several shots from different sources into a single still or sequence is at the heart of all special visual effects, and also of many avant-garde artistic movements—such as Surrealism—that seek to subvert our notions of reality. Images are usually composited with masks (described earlier in this chapter), but when images are composited without masks the process is called image blending.

Traditional Matting Techniques

Before computers became popular production tools, painting, and photographic techniques were used for matting and compositing. Most mattes were created by painting them directly on glass or on film, and compositing was primarily done with the **optical printer**, a camera that photographs film projected onto the camera by a projector mounted in front of it. The optical printer was developed in the early 1940s, and it transformed the way special effects in movies were created for decades. The optical printer can duplicate an entire film onto a new roll, and it can also be used for compositing, slowing, or reversing the motion, reshooting through anamorphic lenses, balancing the contrast and the color values, zooming, panning, and creating transition effects. Today the optical printer is still used in conjunction with digital and high-resolution video technology, but it has been largely replaced by digital compositing, now the primary form of dynamic compositing.

The technique of **matte painting** was developed in the 1920s by making detailed paintings on glass but leaving some areas empty so that the live action can be matted, or inserted, there. This is the simplest kind of matting and it is called a **stationary matte.** Initially the partially painted glass was placed in front of the camera, far enough away that both the painting and the live action could be

14.3.1 A live-action image of the singer composited with renderings of a virtual environment. (From the music video *Agolo* by Angelique Kidjo. Courtesy of Telecreateurs, Medialab Paris, Phonogram, Michel Meyer, and ZAPDAN.)



seen through the unpainted areas in the glass, and then they were recorded together. As this technique developed, the clear areas through which the camera photographed the background action were painted black instead of being left transparent. This way it became possible to add the live action later by rewinding the film and making a double exposure. This innovation made possible the matting of figures in the foreground of the scene.

A **traveling matte** is a dynamic matte that was devised to composite moving elements in the scene. In this case, the blocked areas change every frame in exact synchronization with the foreground image action. The technique of traveling mattes has been perfected over the years—it was first used in the 1940s and 1950s—and today it has become a staple in the digital production of visual effects. Several methods and techniques are in use today for producing a traveling matte, including blue and green screen, rotoscoping, and camera

14.3.2 The Cloud character in *The Stormriders,* played by Aaron Kwok, has the martial craft of pulling water from the surroundings and turning it into a deadly weapon. The liquid attack was created with blobby surfaces in Softimage, and composited onto the live action with transparency, reflection, and refraction masks. (Images courtesy of Centro Digital Pictures Ltd.)

14.3.3 The different components and props of the main character in the feature film *Stuart Little* were rendered in separate passes. Shown here are a wireframe version of the model, a simple shading composited onto the background plate to check for scale and perspective, and the fur elements in a single pass. The final composite can be seen in Figure 14.3.4. (Images courtesy of Sony Pictures Imageworks. © 1999 Global Entertainment Productions GmbH & Co. Medien KG. All rights reserved.)

tracking (read Chapter 13 for more information on all these techniques). A **garbage matte**, which can be stationary or traveling, is used to isolate and remove elements that are not a part of the shot, such as scaffoldings, lighting reflectors, and wires.

Rear and front projection, as well as in-camera compositing, are compositing techniques that do not involve the creation of a mask or matte because the actors or foreground objects function themselves as masks. While very popular during the earlier days of special effects, these three matting techniques are rarely used today since their digital counterparts offer greater control and are usually more efficient. In both rear and front projection, the matting is achieved by recording the actors or objects to be matted over live projections of the background. In **rear projection**, a translucent screen behind the actor or model is used to project previously shot material. While the actor acts or the model is moved the scene is photographed by a camera situated in front of the action. This simple trick was devised in the early 1930s, and it is a common technique still used in the filming of dialogues between actors inside mock-up cars. Street scenes are rear-projected, while the fake car is moved to simulate the vibration produced by motion. When color photography became the standard of the motion picture industry, rear projection became more challenging from the technical point of view. This was because the large amounts of light that were needed to illuminate the scene brightly enough for the slow color film of the time also washed out the images projected on the rear translucent screen behind the live action. So front projection was developed, and in 1968 it was used with excellent results in the milestone film *2001: A Space Odyssey* to composite the image of a human actor in an ape costume with images of real apes shot elsewhere.

Front projection is based on a projector aligned at a 90-degree angle with the camera and a half-silvered mirror aligned at a 45-degree angle in relation to the camera and the projector. A front projection screen made of a highly reflective material is placed in front of the camera, and the actor or model is positioned in between the screen and the camera. The image is projected on the half-silvered mirror positioned at a 45-degree angle, which sends it to the screen. Ninety-five percent of the image bounces back from the reflective screen, passes through the two-way mirror, and goes directly to the camera.

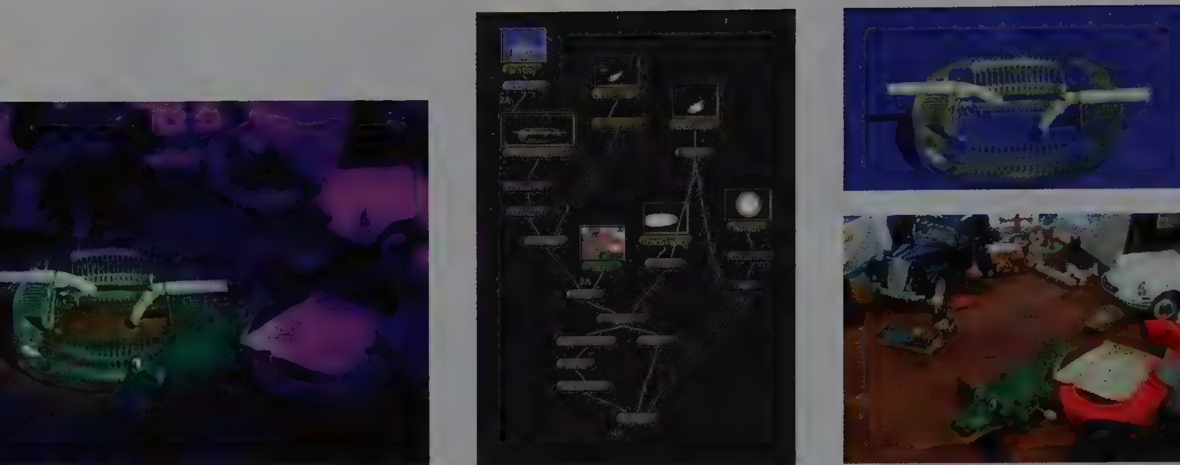
In addition to the optical printer, simpler compositing was also created inside of the camera with a variety of techniques like exposing the same strip of film twice; this was called **in-camera compositing**. The idea of compositing foreground elements over a background is also a staple concept in traditional keyframe animation: motion is simulated with two-dimensional foreground shapes that move over a background. The image compositing in this instance can be done by placing the foreground elements on a cel over the background, or with multiple exposures (sometimes with masks and sometimes without) on the same strip of film.

Compositing with Masks and Operators

14.3.4 Several layers of computer animation and live action were composited and retouched into this shot of the feature film *Stuart Little*. (Images courtesy of Sony Pictures Imageworks.)

A mask used in the compositing process consists of a monochromatic image that protects or masks portions of another image being composited. As described earlier in this chapter, a mask is like a stencil: its solid areas protect the surface being masked, and its perforated areas expose the surface being masked. In many instances the mask or masks used in the compositing process are kept in a separate alpha channel, which is independent from each of the red, green, and blue channels in an RGB color image (Fig. 6.2.4). Image compositing with masks allows us to seamlessly isolate and consolidate multiple images; this is useful when reassembling complex three-dimensional scenes that were rendered in separate parts or layers. Large or complex three-dimensional environments can be rendered in parts, and the resulting images can be assembled back together in two dimensions with image compositing (Figs. 14.3.1–14.3.4).

In addition to compositing with several masks, **operators** or functions can also be used to composite multiple elements. Operators allow users to apply different functions to select parts of the image. Several compositing programs allow users to fine-tune the parameters and to customize their own functions. Figures 14.3.5 and 14.3.6 show two examples that have been composited with a variety of operators. The screen shots show the composite image, the source images, images in process, and the the **node tree** or visual flowchart used to structure the operators and the sequence in which they will be applied to the images. Figure 14.3.5 shows an image with transparencies (glass and smoke) shot on blue screen and composited



14.3.5 The node tree on the right contains the operators that are applied to the two small source images. Three operators are concatenated in the upper right to make the spotlight effect on the ashtray. (RGrad1) is applied to the background to control the color correction, (Mult1) makes the background yellow, and (ColorMatch1) modifies the color. The mask for the (Blur1) operator is made with some ramps combined with the mask coming from (Rotoscope2), and is used to keep the smoke in focus and the front of the ashtray out of focus. (Photo courtesy of Photron. Shake interface courtesy of Apple Computer.)

14.3.6 The visual flowchart (right) shows how a red truck has been added to the original background (thumbnail window, far right). The computer-generated white truck is first inserted, a shadow mask is created, and the truck is colorized. (Photo by Peter Warner. Truck modeled by Caleb Owens. Shake dialog box courtesy of Apple Computer.)

against a desktop. Figure 14.3.6 shows an image manipulation and compositing process that inserts a truck into the original background containing just one car. The node tree shows how the white truck is inserted and placed, colorized, and finally duplicated, scaled, and blurred. Figure 14.3.7 shows an additional example of matting with a green screen, a convenient color for matting because the fleshtones of actors do not have a blue component (read Chapter 13 for more information on blue and green screen details).

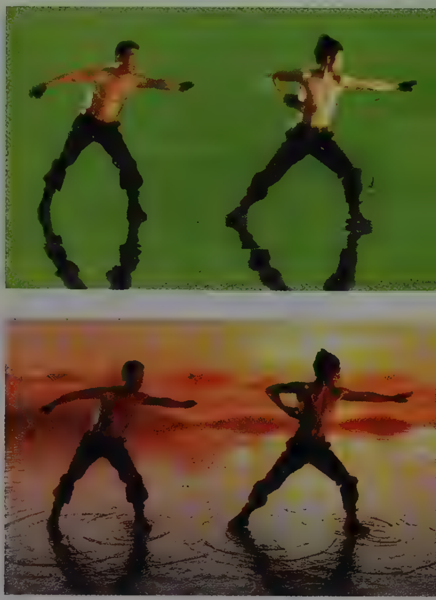
Compositing without Masks

It is also possible to digitally composite images from multiple sources without using masks; in such cases the image compositing process is

usually referred to as **image blending**. With this process multiple images can simply be blended, and the result looks like a collection of translucent, or ghosted, images. Blending can also be controlled with combinations of operators such as addition, difference, overlay, multiplication, and color burn, as illustrated in Figure 14.3.8.

Two-Dimensional Morphing

Renderings of three-dimensional scenes can be modified with two-dimensional interpolation techniques. Two-dimensional morphing is different from the three-dimensional interpolation techniques—sometimes also called morphing techniques—that blend the three-dimensional shape of objects in a simulated scene. **Two-dimensional morphing** is a special type of image blending that interpolates the values of pixels. This interpolation is based not only on the color value of pixels but also controlled by a grid that helps to match and interpolate the shapes of the two images or sequences being morphed (Fig. 14.3.9). The control grid is placed on the two images to be morphed or on two keyframes of the sequence to be morphed. The grid is adjusted in each of the images so that the points on the grid correspond to the areas that have to be morphed. The grid points control the color interpolation of the pixels as well as the spatial interpolation that is necessary for a pixel in the first image to be moved to the XY location of the corresponding pixel in the second image.



14.3.7 These two actors are in a green screen stage. The lighting of the shot is done carefully so that the light does not bounce off the screen and onto the actors. A landscape background is composited after the shot is completed and traveling mattes are extracted off the blue screen. (Images courtesy of Menfond. © StarEast/Bob.)

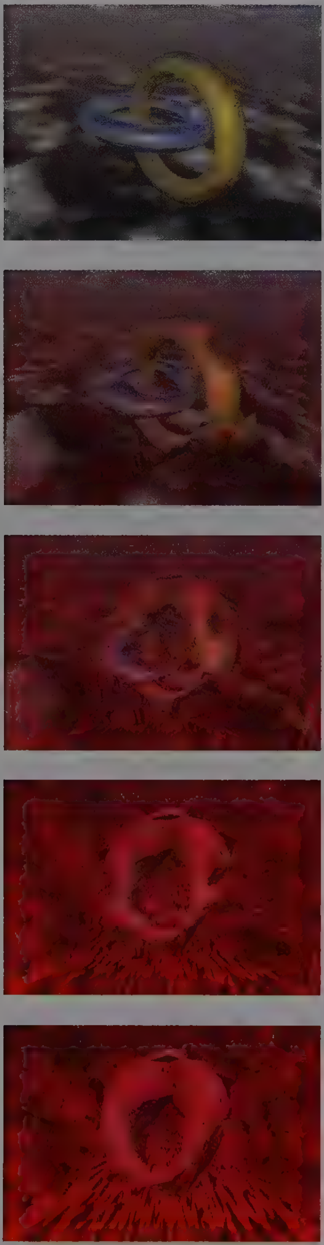
14.4 Image Sequencing

A great deal of a story narrated with images is actually told by the order and timing in which sequences of images are presented to the audience. The arrangement and composition of moving images is called **image sequencing** or **image editing**. The stage of image sequencing in any computer animation project is an important moment in the production process for two reasons. It is when ideas presented in the original storyboard can be finalized in a faithful and flawless way. Image sequencing is also the moment in the creative process when ideas can be fine-tuned to make the project more expressive or to conform to unexpected changes, such as a shorter or a longer running time.

In most computer animation productions, the image sequencing process is usually not done by the individual or production team that produced the computer animation in the first place. The final sequencing of images—and their subsequent output onto film, video, or photographic media—involves specialized techniques and skills. But increasingly, as more of the final image sequencing process migrates to the digital realm, it is common for those involved in producing the computer animation to also be involved in some or all of the image sequencing.

14.3.8 The foreground and background images from Figure 14.1.9 composited without a mask, by applying just the Red Channel of the jungle to the floating rock using the Color Burn operator.

The most common reason for computer animators to get involved with image sequencing is to create an animatic or a rough



14.3.9 Two-dimensional morphing is a technique that blends still images into one another. The blending of the pixel color values is controlled by a grid that tags areas of the image to be blended with one another.

cut of the project. An **animatic** is a preliminary version of a computer animation and is used to visualize how the final project may look (read Chapter 2 for more information on animatics). Unlike an animatic, a **rough cut** of a computer animation usually contains the finished renderings and final motion. However, the final arrangement of the sequences in a rough cut is yet to be locked, and the transitions between sequences are not implemented. Rough cuts are used to preview the rhythm and timing of the animated sequences, and they often result in modifications to the original plan in terms of the duration of a particular shot, scene, or sequence, or in terms of the order in which elements are presented.

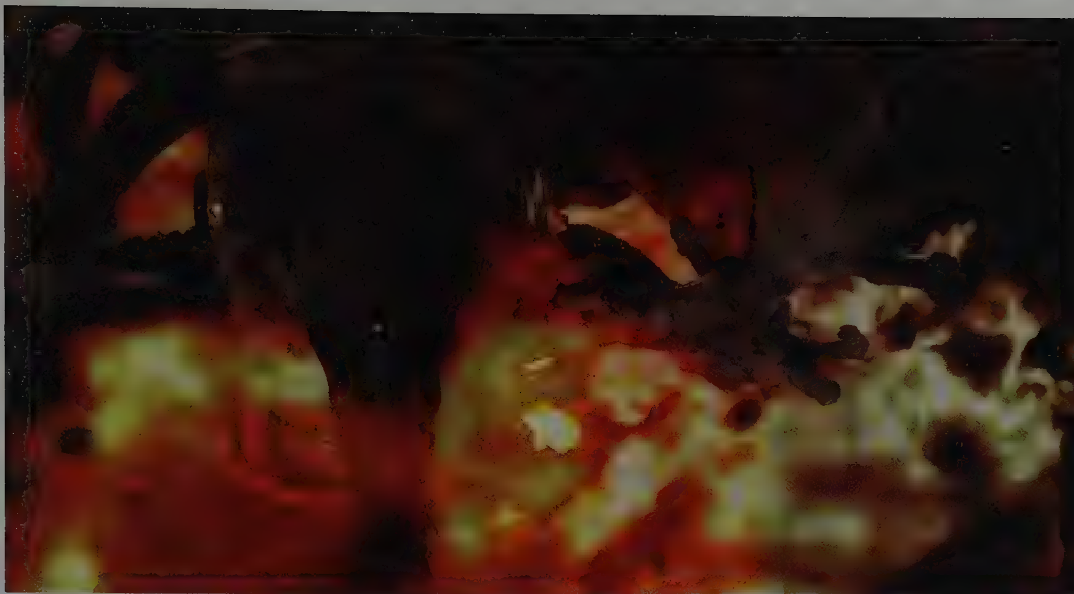
Building a Sequence of Images

Most image sequencing software provides at least one pair of channels or digital **video tracks** or layers for placing the different shots that are being edited as well as the transition effects that link them. The idea of using tracks to build a sequence of dynamic information comes from the predigital worlds of video and sound editing. Since its predigital days sound editing used multiple tracks of sound to layer and composite simultaneous sounds—such as dialogue, background music, and sound effects—from a variety of sources. Most image sequencing software today permits the user to sequence and combine images from any number of tracks, and in any order. This is commonly called **nonlinear editing**, and it makes reference to the fact that digital editing systems do not have to scan a linear medium (such as film or videotape) in order to find the source images. The standard editing technique in predigital film and video, the **A/B roll editing** technique, is based on building a sequence by combining images arranged linearly in two different sources.

Sequencing the different shots in a computer animation is usually done with digital editing software that allows the editor to arrange animated segments by manipulating icons. This is done by dragging the icons that represent the source images into the proper position on the track and dropping them into place. Their location and duration can be further adjusted by sliding them on the video track or time line. This common method of image sequencing is called **copy-and-paste editing** because images are copied from the source files and then pasted in the sequence being built. In addition to the tracks that hold the animation clips, there are tracks for the transition effects between shots placed on different video tracks, and tracks for sound information.

Visual Rhythm and Tempo

Two of the key principles of dynamic image composition include rhythm and tempo (a thorough examination of *all* the principles of dynamic image composition is beyond the scope of this book). In



practical terms, the visual rhythm sets the pace of a computer animation, and tempo sets the speed. In the context of image sequencing, the **visual rhythm** of a sequence or an animated project is the visual pattern created by the frequency of transitions between shots. The rhythm of image sequences can be, for example, predictable or surprising, regular and soothing, syncopated and lively, or irregular and chaotic. The tempo of image sequences ranges from slow to very fast. **Visual tempo**, or visual pace, is set primarily by the length of the individual shots in the sequence. The pace is fast when shots in a sequence stay only a couple of seconds on the screen; the pace is slow when shots are longer in length. When used in conjunction with the action contained in the animated images, rhythm and pace are effective ways of reinforcing the emotional content of a scene.

14.3.10 The compositing in *Final Fantasy: The Spirits Within* averaged 16 three-dimensional layers per shot; the highest count was 498 in a single shot. Explosions were created with a combination of techniques ranging from RenderMan shaders to libraries of practical effects (dust, smoke, and fire) for compositing. (© 2001 FFFP.)

Transitions Between Shots

The techniques used to blend and composite moving images are called **transition effects**. Transitions between shots, as they are also called, can be simple or complex, ranging, for example, from a straight cut to a cross-dissolve combined with a wipe. Image layers, alpha channels, and masks are commonly used to combine sequences of computer animation just as they are used to combine still images.

Transition effects are useful from the visual point of view because they add visual interest and variation to a sequence. Transition effects can add funkiness and ornamentation. In addition to their immediate embellishment functions, transition effects are also an effective way to communicate the passage of time and to anticipate or play down the upcoming action. Transition effects, for

(ReBoot® and © Mainframe Entertainment, Inc. All rights reserved.)

14.4.1 A cut between two animated sequences (far left), and three popular transitions: a fade-out, a cross-dissolve, and a wipe.

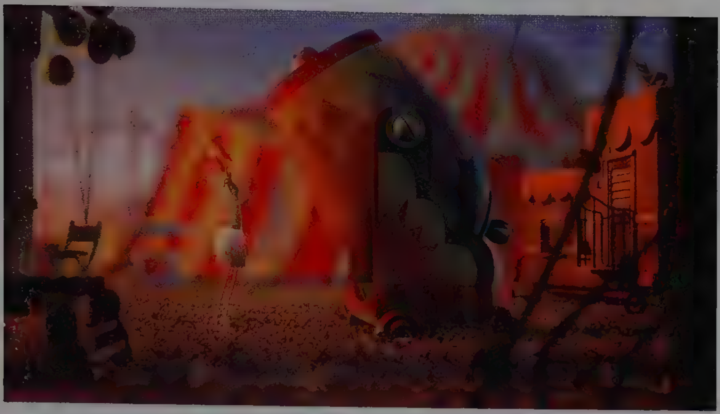
example, break the continuity of one sequence or announce the beginning of a new one.

In addition to the information provided by the storyline and the action shown by the camera, transition effects help to define the **temporal** and **spatial relation** between shots or scenes. A quick and simple transition effect, such as a straight cut, for example, reinforces the fact that the action shown in the second shot of a sequence occurs right after the final shot and also in the same physical location. A long cross-dissolve, on the other hand, hints that two consecutive shots happen in different places and also that a fair amount of time takes place between them. Some of the most commonly used transition effects include the cut, fades, cross-dissolves, wipes, and morphs. (Two-dimensional morphing is covered earlier in the chapter and is illustrated in Figure 14.3.9.)

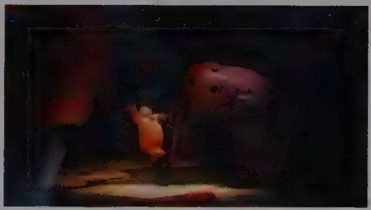
A **cut** is a plain and immediate change from one shot to another. A cut is the simplest transition and also the most common. Cuts are made by placing the last frame of a sequence, or tail, right next to the first frame, or head, of another sequence (Fig. 14.4.1). A cut gets its name from the fact that when editing film a transition of this type is achieved by actually cutting two strips of film and then splicing them together with transparent glue or tape. A **soft cut** is a combination of a cross-dissolve and a cut. This effect provides a cut that is slightly expanded in time, and softened, by the effect of the two shots quickly fading into each other.

In a **fade-out**, the end of a shot vanishes gradually and reveals a still frame of solid color. When a shot fades into a black still frame, the effect is called **fade to black**. Fade-out transitions can also be made between shots so that the end of the first shot gradually vanishes into the early frames of the consecutive shot which suddenly pops into the sequence (Fig. 14.4.1). In a **fade-in** transition—which is the inverse of a fade-out—the first shot usually consists of a still black frame so that the second half of the transition seems to emerge from black. Sometimes a fade-in also starts with a first sequence of frames that is suddenly cut when the second sequence has fully appeared. Fades are defined by their length and by their intensity. Most fades are just a couple of seconds long, but a slow fade can last 10 seconds or more, and a quick fade lasts less than a dozen frames. The intensity of a fade is expressed in percentages. Fades usually go from 0 percent to 100 percent or vice versa, so that when the images are fully faded they are either fully invisible or fully visible. But in some situations partial fades can create interesting effects—for example, a sequence of shots that start their fade-in with a 20 percent fade-in value and are cut when they reach 80 percent.

A **cross-dissolve** is a transition effect where two shots fade into each other; as the first shot fades out, the second shot fades in (Fig. 14.4.1). Cross-dissolves are an effective way to give the audience a moment to pause and think about what they have just seen or what they are about to see. Cross-dissolves are effective links between two shots and can be used as adverbs in the visual grammar of computer



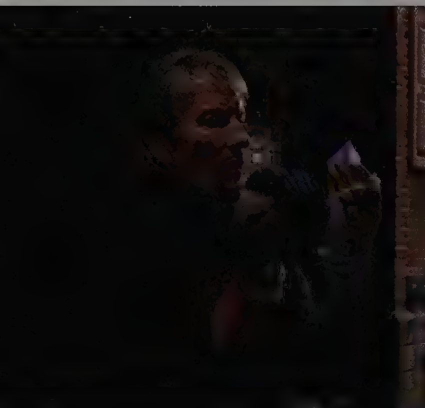
14.5.1 The luscious color look of *Engel zu Fuss* was achieved during compositing, with Combustion software, through color grading and image filtering. The project was initially rendered in layers, with Mental Ray, to facilitate compositing. Rendering passes included beauty, ambient, occlusion, depth, and a few masks. (Image courtesy of Studio Soi.)

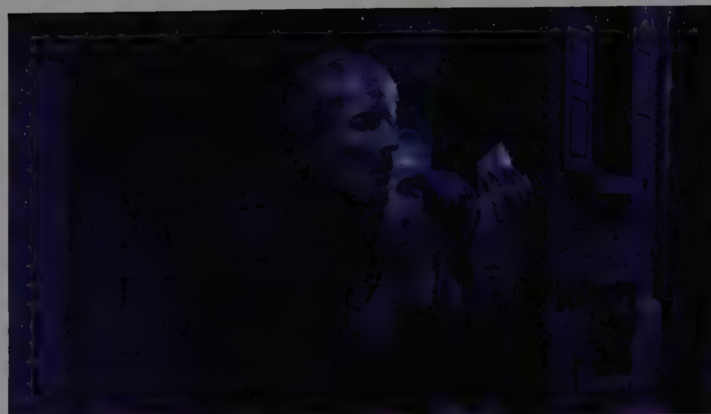


animation. A cross-dissolve can say, for example, "and then," or "in the meantime," or "years later." Cross-dissolve transitions are effective ways to connect actions that happen in the distant past or in the future, or to present actions that occur only in the imagination of a protagonist in parallel with the real action.

There are many variations of cross-dissolve transitions. The most common form of a cross-dissolve consists of a gradual and delicate simultaneous transition from one shot to another. The length and intensity of cross-dissolves is expressed—as it is expressed in fades—in terms of frames and percentages. A very short cross-dissolve is often called a soft cut and is barely perceptible, but it adds an accent of slowness or acceleration to the transition between two shots. Slow cross-dissolves create a ghostly effect where the objects and characters in the scene seem to be transparent. Blending and **layering moving images** is best achieved with cross-dissolves. A dream sequence, for example, where images overlap with one another and objects suddenly materialize or vanish is a classic example of using cross-dissolve transitions for layering images. A **dither cross-dissolve** uses a coarse pattern to fade the two shots into one another.

Wipes are transition effects where the basic idea is that the sec-





14.5.2 The night-for-day look (top right) is achieved by drastically changing the overall color grading of the original live-action shot. (Images courtesy of da Vinci Systems.)

ond shot displaces the first shot by sliding into the frame, dropping or spreading over it. There are myriad variations of wipes, and one is illustrated in Figure 14.4.1. In the most common form of a wipe, called plainly a **wipe** or a side wipe, the second shot slides across the frame over the first shot. Wipes can also slide diagonally, along the vertical axis, radially, or using any geometric shape or edge as a template. Elaborate and stylized wipes include screen splits, spin wipes, interlaced wipes like venetian blinds, and page turns.

14.5 Color Grading

As mentioned in Chapter 13, the color of the images in one or several sequences can be adjusted at different points during the production or the postproduction stages. This color adjustment is called **color grading**, and it is also referred to as color timing, color balance, and color correction. The individuals most involved with hands-on color timing in a live-action movie are usually the colorist, the cinematographer, and the visual effects supervisor. The color supervisor, the lighting supervisor, or the art director usually perform the same task in an animated movie.

Traditionally, color grading of film was done through photochemical processes and it was mostly used to correct mistakes of exposure, lighting, or film stock. **Color timing** derives its name from the fact that film could be left in different chemical solutions for different amounts of time and at different temperatures to achieve different results. Photochemical color timing was, and continues to be, a process that is sometimes difficult to control and where predictability and efficiency run at the expense of experimentation. Digital color grading is fully interactive, offers greater flexibility, and has a greater set of tools than photochemical timing. In spite of its relative newness, digital color grading is widely available and is replacing photochemical grading as the premiere color grading method.

At its simplest color grading is essentially the manipulation of the color and brightness values in a scene. Color corrections can be

applied in layers or in sets: a **primary color correction** may take care of most of the desired result while a secondary color correction can be used to fine-tune details. At its most complex color timing is a collection of filters and functions that control every aspect of the image, and that can be used to transform the overall **visual look** of a scene or an entire movie (Figs. 14.5.1 and 14.5.3). The 2000 movie ***O Brother Where Art Thou?*** by the Coen brothers or ***Le fabuleux destin d'Amélie Poulain*** directed by Jean-Pierre Jeunet (2001), or Peter Jackson's ***The Lord of the Rings: The Two Towers*** (2002), for example, are some of the earliest examples of digital color grading used to drastically transform the overall look of the original live action and visual effects footage. Later examples include the 2005 movie ***Sin City*** directed by Frank Miller and Robert Rodriguez, ***300*** directed in 2006 by Zack Snyder, and ***Speed Racer*** (2008) directed by the Wachowski brothers.

14.5.3 Notice the difference after color grading between the final color hues (top left) and the original blue screen footage (top right). *Fearless* was shot on film and composited with Adobe After Effects software. (Courtesy of Menfond Electronic Art & Computer Design Co. Ltd.)

Many of the tools described earlier in this chapter are available in digital color grading systems: parameter curves, histograms, and selection or masking tools. The most sophisticated selection tools are called **power windows** because of their wide range of functionality. Power windows may select multiple areas at a time and offer a wide range of functionality, including ramps, selected areas and effects that change through time. Temporal control allows color graders to re-time the color of visual elements throughout an entire sequence in a semi-automated way. The hard edge shadows cast by a walking character, for example, could be first selected throughout the entire sequence, made lighter and slightly warmer in color, and given softer edges. One of the most dramatic uses of color grading is the **night-for-day** effect in which typically a scene shot in daylight is manipulated to the point of transforming the entire color palette and simulating the color and lighting conditions of a night scene (Fig. 14.5.2).

Chapter 14

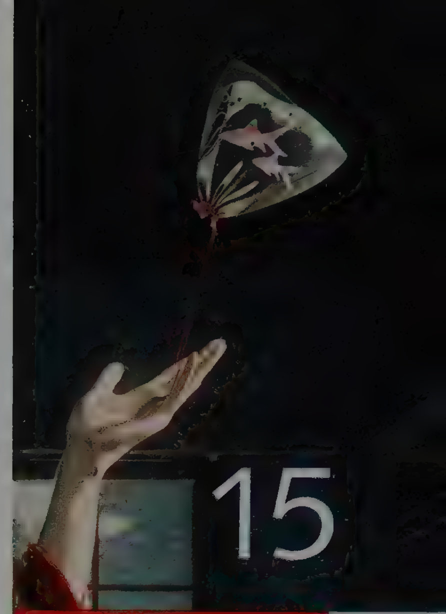
Key Terms

A/B roll editing
Alpha channel
Angular straight lines
Animatic
Background
Blurring filters
Brightness
Channels
Collage
Color dithering
Color grading
Color look-up table
Color timing
Contrast
Copy-and-paste editing
Cross-dissolve
Custom filter
Cut
Darkroom
Deinterlacing filters
Digital filters
Digital retouching
Dither cross-dissolve
Edge detection filters
Fade to black
Fade-in
Fade-out
File size
Filter radius
Foreground elements
Front projection
Garbage matte
Highlight
Histogram
Histogram editor
Image blending
Image dimensions
Image editing
Image layers
Image manipulation
Image sequencing
In-camera compositing
Indexed color
Interpolation techniques
Layering moving images
Linear color space
Log color space
Mask, masking
Matte, matting
Matte painting
Midtone
Neighbor pixel interpolation
Night-for-day
Node tree
Nonlinear editing
Nonlinear color space
NTSC color filter
Operators
Optical printer
Parameter curves
Photographic procedures
Pixels per inch
Postprocessing
Postproduction
Power windows
Primary color correction
Rear projection
Resampling an image, the color resolution
Retouching
Rough cut
Shadow
Sharpening filters
Soft curves
Soft cut
Spatial relation
Stationary matte
Stencils
Straight diagonal line
Studio
System palette
Temporal relation
Texture noise
Tonal range
Traditional painting tools
Transition effects
Traveling matte
Two-dimensional morphing
Video tracks
Visual tempo
Visual flowchart
Visual tempo
Weighted interpolation
Wipe

(A composited computer-generated train. *Negotiator: Mashita Masayoshi.* © 2005 Fuji Television/ ROBOT/TOHO/ SPWT.)

CHAPTER 15

Image Resolution and Output



QUICK INDEX

(Top: Detail of *Geisha with Goldfish*. © Keica, Inc.)

Summary

THE BASIC CONCEPTS REQUIRED TO OUTPUT computer-generated images in a variety of media are the focus of this chapter. The three types of image resolution—color, spatial, and temporal—are examined in detail. File formats are reviewed, as are the most popular delivery media, including video, film, paper, and CD-ROM, and their aspect ratios.

15.1 Basic Concepts of Digital Output

Each of the professions and industries in which three-dimensional computer imaging is used today—for example, live action movies, TV animation, and platform games—has different output requirements. This is due to the differences in the final product and the different forms of final delivery and distribution of products in each of these areas. The output of three-dimensional imagery in many of the art and design areas that deal with two-dimensional creation, for example, occurs primarily in the form of paper printouts and film slides, or transparencies. These areas include, for example, illustration, photography, and graphic arts. The new areas of interactive art, entertainment, and virtual reality that include interactive videogames and online information services, for example, require that images be delivered in a variety of digital file formats. This delivery often happens through computer networks, or on magnetic or optical media. In some of the three-dimensional areas, including product design, sculpture, and architecture, the digital output includes both printouts for presentation purposes and digital files for computer-aided design and computer-aided manufacturing (CAD/CAM). In four-dimensional, or time-based, activities—including computer animation, gaming, and location-based entertainment—delivery usually takes place in the form of video, film, or digital files.

The digital output process starts when rendered images are taken out of the computer system with the **output peripherals**. These include an array of devices such as printers, pen plotters, film

or video recorders, and three-dimensional milling or casting machines (Fig. 15.1.1). Output peripherals are used for fixing the images created on the monitor onto other media that may include paper, film, video, and CD-ROM. This capture process is not automatic. It is based on a **translation of data**. This translation is made by software and electronic components of output peripherals called **digital-to-analog** converters or **DTAs**. The DTAs convert the digital information created with software back into continuous information. This process is just the inverse of scanning an image on paper that is to be used as a texture map or a background in a three-dimensional scene. In this case, the sensors in the input peripherals—called analog-to-digital converters—convert the continuous information contained in the image into digital information that can be manipulated by the program. DTAs convert the binary numbers that describe an image back into analog voltages that are subsequently converted into light, heat, or pressure by the imaging components of the output peripherals. The quality and sophistication of the digital-to-analog converters—their precision, definition and speed—often defines the quality of the final output as well as the cost of the output peripheral. High-quality DTAs provide digital output with low noise, high image resolution, and a wide chromatic range.

15.1.1 The body parts for the *Medicus* project were machined from a block of dark blue jewelers wax using the Roland MDX-20 milling machine. Due to the limitations of a three-axis output device, the model is created in several sections to have access undercuts. See Figure 15.8.2 for a later step in the fabrication process. (© 2002 Dan Platt, Solid Image Arts, LLC.)

15.2 Image Resolution

Image resolution can be defined as the amount of detail contained in an image or sequence of images. The resolution of an image is also called **image definition**, and it is related to many factors such as the quality of the input and output peripherals, the color depth of the computer system, the capabilities of the software that was used to create the image, and the quality of the output media. Having a good understanding of the basic issues of image resolution is fundamental in order to use three-dimensional computer imaging techniques to their fullest. There are four aspects of image resolution that are relevant to visual creators: spatial resolution, chromatic resolution, temporal resolution, and image compression.

Spatial Resolution

Computer-generated images are made of **pixels,** or picture elements, which are the little dots that we see when we get very close to the computer screen. **Spatial resolution** describes the amount of image detail and definition in digital images. Spatial resolution, also known as **pixel resolution**, has to do with the total number of pixels in an image, and is defined by the relation between the dimensions of an image and the number of pixels in the image. Spatial resolution can be expressed in terms of pixels, dots, or lines per inch.

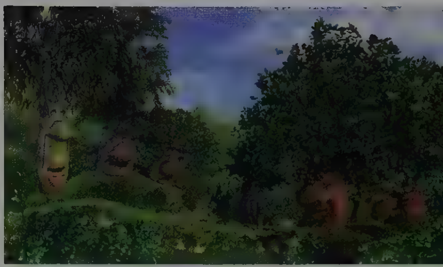
The number of pixels that exist in an **image file** is measured in **pixels per inch** or **ppi**. When rendering a three-dimensional scene, for example, it is necessary to indicate the pixel resolution at which

the image is to be rendered. The most general way to specify the absolute spatial resolution of a file is by using pixels per inch, regardless of the resolution of the output peripheral used to print or display the image. The spatial resolution of image files ranges from a low of less than 100 ppi to a high of 4,000 ppi.

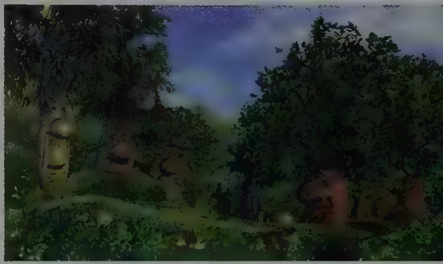
The spatial resolution of a specific input or output **peripheral device** can be measured in **dots per inch** or **dpi**. This unit of resolution is often related to the number of sensors in an input peripheral, or the number of imaging heads in an output peripheral. A medium resolution ranges from 300 to 600 dpi, and high resolution ranges from 600 dpi and up.

Lines per inch or **lpi** is another unit for measuring spatial resolutions. lpi is used almost exclusively in digital prepress or in desktop publishing for measuring the number of lines—or rows of dots—in **halftone screens**. These screens are used for printing an image with traditional graphic arts mechanical reproduction techniques. Halftone screens of 65 lpi, for example, are commonly used for newspaper image quality while 150 lpi halftone screens yield great image definition on high-quality coated paper. A simple formula that can help determine the pixel (ppi) resolution needed when preparing files for **halftone output** specifies that the ppi resolution should not exceed 2.5 times the target lpi resolution (2.5:1 ratio). For example, a resolution of 300 ppi would be adequate when using halftone screens of 150 lpi (2:1 ratio) but probably overkill if 65 lpi screens (4.6:1 ratio) were used. The top image in Figure 15.2.1 is a 2K file at a spatial resolution of 600 pixels per inch, which has been resampled down to show the visual effects of lower spatial resolutions. A **2K file** is a generic name for files with dimensions of 2048 pixels wide and above 1080 pixels high. A standard HD file, at 1920 × 1080 pixels, is almost the size of a 2K file. The pixel width of 2K files is usually fixed but the height varies depending on the aspect ratio of the frame. A 35 mm Full Screen 1.33 Aspect Ratio 2K frame, for example, is 2048 × 1536 pixels (3.14 Megapixels) while a Super 35 mm 2.4 Aspect Ratio 2K frame is 2048 × 854 pixels (1.74 Megapixels). Big difference—one file is almost double the size of the other. Likewise with **4K files**, the generic name for files that are 4096 pixels wide. A frame of 35mm Full Screen 1.33 Aspect Ratio at 4K, for example, is 4096 × 3072 pixels (12.58 Megapixels) while a frame of Super 35 mm 2.4 Aspect Ratio at 4K is 4096 × 1708 pixels (6.99 Megapixels).

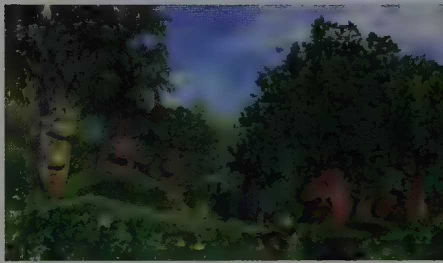
When the **dot resolution** of the output peripheral matches exactly the pixel resolution of an image file, we can see the contents of the file displayed in optimal fashion, but these cases are rare. Most often we work with files whose ppi resolution does not match the dpi resolution of an output peripheral. In those cases, we can use specialized software to **resample** the pixel resolution of the file and match it to the dot resolution of the output peripheral. We can also let the internal software of the peripheral do the resampling for us. However, the results are generally better and more predictable when we do it.



600 PPI RESOLUTION



200 PPI RESOLUTION



75 PPI RESOLUTION

30 PPI RESOLUTION

15.2.1 Four different spatial resolutions (ranging from 30 to 600 ppi) reproduced here at the same 150 lpi halftone screen resolution. The top image is the original rendering at 2K resolution. (© 2001 Dygra Films.)

15.2.2 This sequence shows the 300 dpi images from Figure 9.6.9 output at the lower resolution of 72 dpi.

Using pixels per inch as the unit to convey spatial resolution is convenient because often the resolution of the rendered images is different from the resolution of the peripherals used to output them. Images may be rendered at one resolution and displayed at another. For example, when an image is rendered in high resolution and then displayed in high or in low resolution, the high resolution image loses detail when displayed on a low resolution display, but it shows its true resolution when shown on a display with a higher resolution (Fig. 15.2.2). The fact that an image may look low resolution on a specific printout or monitor does not necessarily mean that the image is low resolution.

For example, a three-dimensional scene rendered at a resolution of 300 dpi, and horizontal and vertical dimensions of 3×2 in., has an absolute pixel size of 900×600 pixels. On a monitor with a resolution of 300 dpi, for example, there would be a 1:1 match so that each pixel in the file corresponds to a dot on the monitor. On a monitor with a resolution of 100 dpi the results would be different due to the 3:1 ratio, 3 pixels for every dot. In cases such as this one, the discrepancy can be solved without resampling the file by keeping either the resolution or the dimensions constant. If the resolution is kept constant, each pixel in the image file is assigned to one dot in the output peripheral. In this case, the resolution is untouched, but the physical dimensions change. In the example of the 300 dpi 3×2 in. file the new dimensions at 100 dpi would be 9×6 in. If the dimensions are kept constant, every three pixels in the image file are assigned to one dot in the output peripheral. This leads to a significant loss of spatial resolution, but the dimensions remained untouched. The 300 dpi 3×2 in. file remains at the same size but with a lower resolution.

Spatial resolution is also used to define the amount of detail in video (not RGB) images. The number of horizontal lines of resolution is used to measure the spatial resolution of video. Some of the most common resolutions are 525 lines for the NTSC format, 625 for both the PAL and SECAM formats, and 720p and 1080i for the still-evolving HD formats (720 lines for the progressive version and 1,080 for the interlaced version). As video technology and computer technology get closer together we might end up using pixels as the single unit for defining spatial resolution. (The section in this chapter on temporal resolution contains more detail about video formats.)

Color Resolution

The **color resolution** of an image is related to the amount of colors and shades of gray that may be contained in it. Color, or chromatic, resolution is determined by the number of bitplanes used to create and display that image. A **bitplane** can be described as a grid where each cell in the bitplane stores a one-digit number. Since each cell on the grid is assigned to a **pixel**, or picture element, on the computer screen, the numerical value is translated into the color that is displayed on the pixel controlled by that cell. Multiple bitplanes can be

thought of as layers in the **graphics memory** or **bitmap**. It is the number of layers in a bitmap that determines the number of colors that each pixel on the screen may have. Because of this, color resolution, or **color depth** as it is also called, is easily defined by the number of bits used to define each channel of an RGB color—for example, 8-bit, 10-bit, 12-bit, 16-bit, 24-bit, or 32-bit color.

A pixel in a bitmap with just one plane, for example, is capable of displaying only one of two colors: black or white. This is because each cell in a one-level bitmap grid can only store a one-digit number, in this case a zero or a one. When a bitmap has more than one plane then the numbers in the corresponding cells of each plane are added—using the conventions of the binary numerical system—to make a longer value. A bitmap with two planes is capable of displaying one of four possible colors—or values of gray—in each pixel (Fig. 13.1.4). This is because when we add the ones or zeros in the corresponding cells of each plane we end up with four possibilities: 00, 01, 10, and 11. In the binary numerical system, 00 equals our decimal 0. Binary 01 equals decimal 1, 10 equals 2, and 11 equals 3.

A bitmap with three planes is capable of displaying one of eight colors per pixel (Fig. 15.2.3). When we add the binary values (1 or 0) in the corresponding cells of each bitplane we end up with eight possible values expressed here, first in binary and then in decimal form.

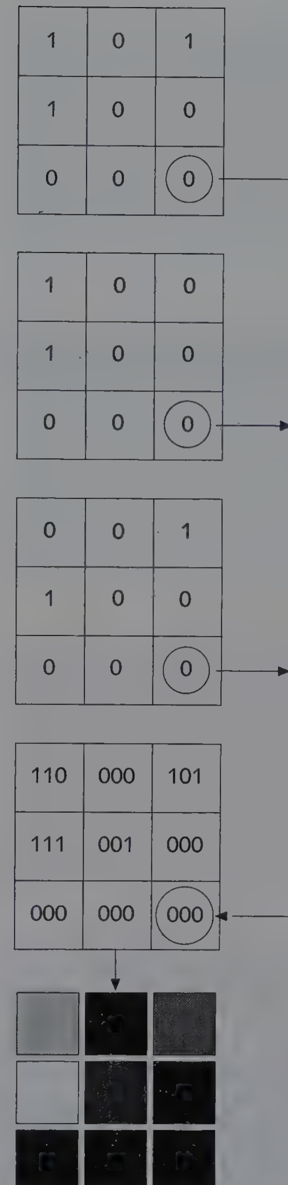
000 = 0	011 = 3	110 = 6	001 = 1
100 = 4	111 = 7	010 = 2	101 = 5

In the **binary system**, numerical values are read from right to left. The digits in the first (rightmost) column represent the units, and their decimal value can be found by multiplying them by the number one, which is the result of two to the power of zero (2^0). The digits in the second column are multiplied by two, which is the result of 2^1. The digits in the third column are multiplied by 2^2.

BINARY	$2^2 = 4$	$2^1 = 2$	$2^0 = 1$	
DECIMAL	$\times 1$	$\times 1$	$\times 0$	
	4	**+ 2**	**+ 0**	**= 6**

The rule is very simple: the more bitplanes in a bitmap, the more color may be displayed in an image. The maximum number of colors that may exist in a computer-generated image can be calculated by elevating the number two—the base number in the binary system—to the power of the number of bitplanes. Standard configurations of color resolution include 8-bit color with 256 colors, 16-bit color with 65,000-plus colors, and 24-bit color with 16 million-plus colors. Figure 15.2.4 shows a list of several possible configurations.

In addition to the color resolution of an output peripheral, it is also important to consider its chromatic range, and the color conversion techniques used to convert color values from one color space to another. The **chromatic range** of any device defines the range of



15.2.3 A three-level bitmap can display up to eight colors. The values of each of the bitplanes are added to determine the final value of the pixels.

colors in the visible spectrum that the device is capable of reproducing. Not all media and techniques for creating color are capable of creating exactly the same colors. The CIE chromaticity diagram (Fig. 6.2.3) is a useful tool for visualizing the chromatic ranges of different media. This diagram can be used to bridge the different ranges or gamuts of color that are obtained when different color systems are used. It can also be used to narrow the amount of color that gets lost when an image file is output. Knowing which colors overlap across different media helps to work around the physical limitations of color reproduction.

The fact that many of the colors created with RGB computer monitors are too bright and saturated for display on standard television sets illustrates a limitation created by the different chromatic ranges of different media. Many of these RGB colors have to be clipped before the image is transferred to videotape; otherwise, these saturated colors would fall outside the chromatic range of video. Clipping the colors that fall outside a chromatic range does not mean that the colors are removed altogether. Instead, they are replaced with a color that is within the chromatic range and that resembles it the most. **Color clipping** is useful for avoiding distortions, such as color bleeding, in the final video recording. **Color bleeding** is the streaking and overflowing of colors that occurs on a television set when the colors in the video signal are too saturated. (To create color bleeding, turn the saturation controls on your television set all the way up.)

It is often necessary to **convert colors** from one color space to another—for example, from the RGB color space to CMYK, as we convert an image rendered and retouched in the RGB color space into a **CMYK four-color separation** suitable for reproduction in a magazine. Each of the colors in an image is separated into its CMYK components because the printing presses used in mechanical reproduction reproduce color images by printing each of the CMYK layers with a different plate and ink (Figs. 6.2.1 and 6.2.5). In most cases, this color conversion and separation is done automatically by the computer software, and the quality of this conversion varies from program to program. In general, this conversion takes into account many production details—such as type of paper, ink, and printing press characteristics—related to the final printing of an image with a CMYK medium. The color conversion software will use default values if those details are unknown at the time of conversion.

The color depth of a device becomes important when digitizing analog images, as is the case when a digital video camera records a live scene or when a datacine film scanner digitizes 35 mm film. When the analog video signal is digitized, for example, each pixel is captured as a numerical value that defines a specific level of color or grayscale. This analog-to-digital conversion process is called **quantization**. As mentioned earlier in this section, the shades of color are more precise when more bits or digits are used to represent a color in the form of a numerical value.

Total Number of Colors per Number of Bitplanes

$2^{(\text{bitplanes})}$ = possible colors

2^1 = 2 colors
2^2 = 4 colors
2^3 = 8 colors
2^4 = 16 colors
2^5 = 32 colors
2^6 = 64 colors
2^7 = 128 colors
2^8 = 256 colors
2^{10} = 1,024 colors
2^{16} = 65,536 colors
2^{20} = 1,048,576 colors
2^{24} = 16,777,216 colors
2^{32} = 4,214,967,296 colors

15.2.4 The total number of colors that a computer can display is based on the number of bitplanes available in its graphics memory. The number 2 raised to the power of the number of bitplanes results in the maximum number of colors that may be displayed by the system.

The **color sampling rate** is another aspect of color resolution that is relevant only when working with component digital video. The color sampling rate determines the frequency at which the analog video information is encoded into its luminance and chrominance components. In a fully digital world this sampling occurs at the full-color rate of 1:1:1, also known as **4:4:4 color**. This means that for every 4 samples of the luminance (Y) component there are also 4 samples each of chrominance (Pr and Pb). Unfortunately, sampling at a 4:4:4 rate requires more bandwidth than is available in most digital video equipment, so compromises are often made to achieve the highest possible color sampling rate at a reasonable cost. The color sampling rate used for NTSC mini-DV, for example, is 4:1:1; for HDV and DVD it is 4:2:0; for Blu-ray it may be 4:2:0, 4:2:2, or 4:4:4. The rate commonly available in 24p HD systems is 4:2:2, although that signal can also be output as low as 3:1:1 when recorded onto HDCAM videotape, or as high as 4:4:4 when recorded directly onto a hard disk RAID array.

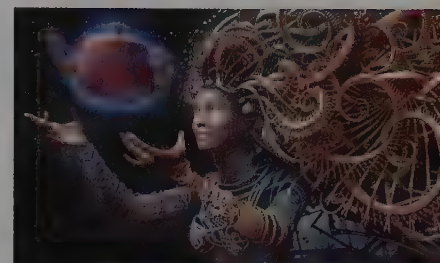
For optimum color results it is important to make sure that the RGB monitor connected to the computer system being used is properly calibrated so that the colors displayed on the screen are as close as possible to the color values contained in the image file. For optimum **color calibration** it is always best to use specialized software and to enlist the help of a color expert. One small but important factor that always has a great impact on the proper calibration of an RGB monitor is the **gamma factor**. This factor is a number used to scale the brightness of the image according to the characteristics of the final output media. When the gamma factor has a value of one, for example, the numerical values of RGB color sent by the computer are used as they are to determine the voltages used by the RGB color guns located inside the monitor to create color. When the gamma factor has a value higher than one, the RGB numerical values are scaled by that number before they are applied to the voltages that control the colors emitted by the monitor (Fig. 15.2.5). A typical value used for video output, for example, is 2.2.

Temporal Resolution

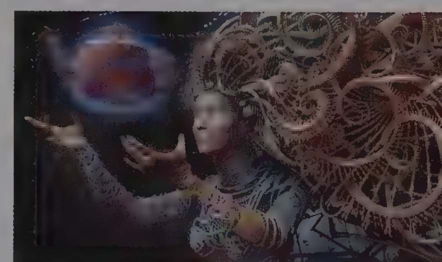
Temporal resolution is related to the number of still images contained in a sequence of images displayed over time. Temporal resolution is almost always measured in terms of the number of images displayed within a specific amount of time, usually a second. Images in a sequence are often called **frames** because of the still frames contained in a strip of photographic film. A common unit for measuring temporal resolution is called **frames per second** or **fps**.

Each output media has a typical temporal resolution. Image sequences displayed on 35 mm film, for example, have a rate of 24 fps (early silent films ran at 16 and 20 fps). Temporal resolution in multimedia projects can range from 8 fps to 60 fps depending on the nature of the sequence, and on the playback power of the computer

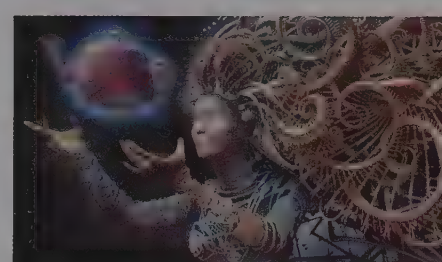
1.0 GAMMA



1.4 GAMMA

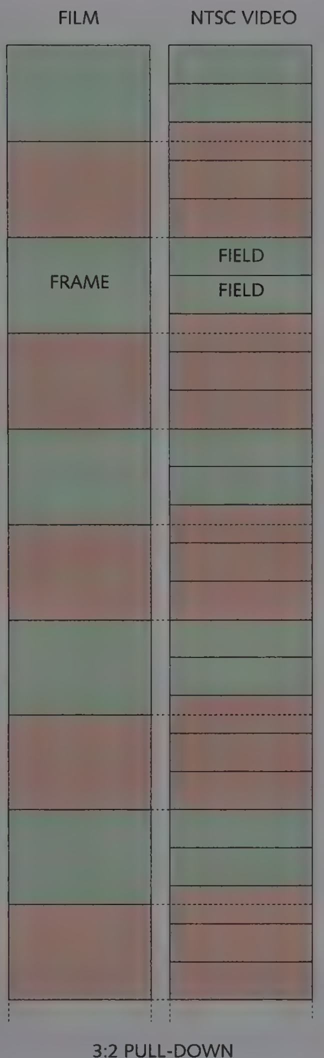


1.8 GAMMA



2.2 GAMMA

15.2.5 Four different gamma values applied to *Animation Mother.* Gamma factors are used to calibrate different devices and media to optimize the display and recording of images. (Image courtesy of Meats Meier.)



15.2.6 The 3:2 pull-down process used to transfer film to video or video to film converts the odd frames of film into two video fields and the even frames of film into three video fields.

system. On **NTSC** video the rate is 30 fps (29.97 to be exact). NTSC is the acronym for the **National Television Systems Committee**, and is the standard used in the United States, Japan, and most of Latin America. RGB purists jokingly refer to NTSC as *Never The Same Color*. The **PAL** and **SECAM** video formats are both displayed at the temporal resolution rate of 25 fps. PAL (**Phase Alternation by Line**) is used in Great Britain, Australia, parts of Europe, Asia, and Africa. SECAM is used in France and some European and African countries.

The conversion of temporal resolutions between film and NTSC video takes place through a standard process called a **3 to 2 pull-down**. This process is based on the fact that each frame of NTSC video is made of two **interlaced fields**. One field contains all the even scan lines in the frame, and the other field contains the odd scan lines. The 3:2 pull-down process converts the odd frames of film into two video fields and all the even film frames into three video fields (Fig. 15.2.6). The video-to-film conversion works by applying the inverse formula.

Image Compression

High-definition full color images contain so much information that it is often impractical to store, transmit, or playback all the bits that represent their spatial, chromatic, and temporal resolutions. **Image compression** techniques are used to minimize the size of image files while preserving as much quality as possible. Image compression is especially important today because of the large volume of files transmitted via public computer networks like the World Wide Web, high-bandwidth private networks, or private intranets.

There are many types of compression techniques and each one of them yields different levels of efficiency and visual accuracy. The programs used to perform the compression and decompression of image files are called **codecs** (compression decompression software). Many of them are integral parts of some of the most popular file formats for moving images like JPEG, MPEG-2, MPEG-4, and QuickTime, and some can compress and decompress the information on the fly. Compression is about reducing the size of the file so that the transmission time is shorter and the storage requirements are smaller. Decompression is about bringing back the data to a format that can be viewed with standard software.

In terms of fidelity to the original file, compression techniques are separated into lossy and lossless. **Lossy compression** techniques discard image information as they compress the file. When a lossy-compressed file, JPEG for example, is decompressed the result differs from the original; the amount of difference will depend on the various user-controlled settings and filters employed in the compression (Fig. 15.3.3). **Lossless compression** techniques, on the other hand, retain the original information as they compress a file. But in spite of their fidelity, some lossless techniques can yield expanded files that may be slightly different from the original file. This is because differ-

ent graphics software, computers, and graphics cards might calculate the floating point decompression in slightly different ways.

15.3 Image File Formats and Aspect Ratios

Files that contain images are called **picture files** or image files. Computer imaging programs can save and retrieve images in a variety of **file formats**. Some of these formats are native to a specific program. This means that the files can only be retrieved by the program that was used to create them, and not by any other program. One solution to the lack of compatibility between **native file formats** consists of saving the visual information in **universal** or **portable image file formats**. Some of these portable file formats include PICT, TIFF, and QuickTime, and are described next. In those few cases when the images cannot be saved in a portable format, it is possible to convert one native file format into the native file format of another program. This alternative solution to the file incompatibility problem is called **file format conversion**, and it is a standard feature in many computer imaging programs in the form of **import** and **export tools**. These tools translate image data files from and into other native or universal file formats. The results obtained with different file conversion utilities vary widely. Some file conversions are almost flawless, while others rarely produce desirable results. There is no easy way to know if a file conversion program will work or not; each has to be tried and evaluated. All file format conversions are directed by so-called **import** and **export filters**, which are tables that instruct the conversion utility how to translate each and all of the elements encountered in the original file.

There are several file formats for both still and moving images. Each of these has been designed with a specific goal in mind and, therefore, is better suited for a particular task. There are always trade-offs between the characteristics of a file; for example, a file format that preserves the fine detail in the image may also take up a lot of storage space (Fig. 15.3.1). It is common in everyday productions to integrate images in different file formats into one single document. It is also quite common to save an image in a variety of file formats as the image is created with different programs, or transferred between different computer platforms through a network (Fig. 6.11.1).

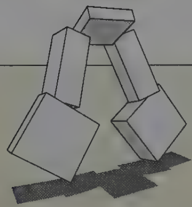
Still Images

Some of the most popular portable file formats for saving two-dimensional visual information in digital form include: PICT, TIFF, EPS, TGA, Cineon, JPEG, BMP, and GIF. There are many other portable file formats for saving still images, but an exhaustive listing is beyond the scope of this book. In addition to the dozens of universal file formats, there are also many native file formats that can only be read by the software that created them. In addition to being able to save single still images these file formats can also save **series of still images**.

File Formats and Storage Requirements

TIFF 16-bit	14.8 MB
TIFF 16-bit LZW	12 MB
PDF high quality	8.7 MB
CIN 10-bit	8 MB
TGA 32-bit	8 MB
BMP 24-bit	6 MB
TGA 24-bit	6 MB
TIFF 8-bit	6 MB
TGA 24-bit RLE	5.3 MB
PICT 32-bit	5.1 MB
BMP 16-bit	4.1 MB
TGA 16-bit	4.1 MB
TIFF 8-bit LZW	2.4 MB
PICT 16-bit	2.2 MB
TGA 16-bit RLE	2.1 MB
JPEG best quality	992 KB
GIF 8-bit	384 KB
JPEG high quality	384 KB
JPEG medium quality	288 KB
JPEG low quality	224 KB

15.3.1 This chart compares the amount of storage required by the same still image (1920 × 1080 pixels, RGB) in a variety of file formats with different color resolutions and compression ratios. The compactness of the file format increases as you go down the list. (*The Missing Lynx*, © Kandor Graphics, S.L./Junta de Andalucía, Consejería de Medio Ambiente/Green Moon España, S.L./ Perro Verde Films, S.L.)



```
%!PS-Adobe-3.0 EPSF-3.0
%%Title: Sims-1.eps
%%BoundingBox: 0 0 200 200
%%EndComments

200 200 scale
.001 setlinewidth
1.5 setmiterlimit
/quad { setgray newpath moveto lineto
lineto lineto closepath
  gsave fill grestore .0 setgray stroke } def

0.4531 0.1716 0.4111 0.0544 0.6423
0.1391 0.6648 0.2546 1 quad
0.6648 0.2546 0.6423 0.1391 0.6456
0.1343 0.6677 0.2502 0.865 quad
0.6997 0.5376 0.7810 0.5643 0.8535
0.9500 0.7641 0.8703 1 quad
0.7073 0.5358 0.7898 0.5622 0.7810
0.5643 0.6997 0.5376 1 quad
0.6456 0.1343 0.6423 0.1391 0.4111
0.0544 0.4145 0.0500 1 quad
0.7392 0.3909 0.6664 0.4162 0.6398
0.2082 0.7161 0.1902 0.975 quad
0.5966 0.2084 0.6724 0.1909 0.7161
0.1902 0.6398 0.2082 1 quad
0.5966 0.2084 0.6398 0.2082 0.6664
0.4162 0.6240 0.4105 1 quad
0.6061 0.4571 0.6747 0.3831 0.6606
0.4622 0.5828 0.5444 1 quad
0.6747 0.3831 0.7774 0.4697 0.7777
0.5593 0.6606 0.4622 0.887 quad
0.7810 0.5643 0.7898 0.5622 0.8620
0.9474 0.8535 0.9500 1 quad
0.6985 0.6377 0.5828 0.5444 0.6606
0.4622 0.7777 0.5593 1 quad
0.6347 0.5379 0.5792 0.5772 0.3892
0.6422 0.4382 0.6031 0.956 quad
0.4382 0.6031 0.4076 0.5317 0.6053
0.4730 0.6347 0.5379 1 quad
0.4382 0.6031 0.3892 0.6422 0.3608
0.5746 0.4076 0.5317 1 quad
0.1447 0.3282 0.2239 0.2544 0.4006
0.5145 0.3501 0.6173 0.831 quad
0.1447 0.3282 0.1380 0.3320 0.2179
0.2579 0.2239 0.2544 1 quad
0.3501 0.6173 0.3436 0.6204 0.1380
0.3320 0.1447 0.3282 1 quad
showpage

%%EndDocument
```

15.3.2 Listing of the EPS program that generated the line drawing of the creature above (described in more detail in Chapter 12).

The majority of computer animations that are eventually recorded on film or videotape are rendered as series of still frames because each frame has to be recorded, one at a time, on film or videotape. Series of numbered still files are commonly used to save the still frames in a computer animation sequence. The numbering of sequential files is usually done automatically by the animation program that creates them. One convention for numbering files in a sequence consists of including the sequence, scene, frame numbers and versions in the filename—for example, Sq011_Sc0079_Fr4099_v037, and so on.

Some of the issues that distinguish one file format from another includes their popularity, their compression schemes, their ability to store alpha or Z-depth channels, and their color depth. Today most still image file formats offer a variety of color depth flavors that range from 8-bit to 16-bit per each (RGB) color channel, or 24-bit to 48-bit for the entire file. Originally most of these portable file formats could only save color with a resolution of 8 bits of color for each RGB channel. Two notable early exceptions to this standard include TIFF, which can save color information in 8-bit or 16-bit linear modes, and Cineon, which saves color in 10-bit logarithmic mode. The more recent OpenEXR format offers 16-bit logarithmic color depth.

The **TIFF** file format, from **Tagged Image File Format**, is popular in a variety of production environments ranging from prepress to animation, and is especially useful when the rendered image has to be reproduced in a printed publication. The TIFF format has 8-bit and 16-bit color per channel versions, and it preserves detailed grayscale information that is fundamental for generating the high-quality halftones (grids of dots of varying size) used in the graphic arts. TIFF files tend to be large and several applications usually provide compression options. **LZW**, short for Lempel-Ziv-Welsh, is a popular compression technique developed in 1984 that is commonly applied to images in the TIFF and GIF formats.

The **EPS** file format, or **Encapsulated PostScript**, is also popular in prepress applications, and can be quite useful and effective when high-quality line wireframe drawings are needed. Information saved in the EPS file format is always imaged at the best resolution possible in EPS-compatible output peripherals because EPS is a **device-independent file format**. EPS files usually require significant amounts of memory for storage and transfer. EPS files are almost identical to PostScript files except for the **header information** that is found at the beginning of EPS files. This header information is inserted automatically by the application that generates the EPS file, and it includes data that is needed to import the file into another application program and output it properly (Fig. 15.3.2).

The **Cineon** uncompressed file format is commonly used in the motion pictures and visual effects industries to store images that have been scanned from film. This format was developed by the Kodak Eastman Company to capture the subtlety and dynamic range of images originally shot on motion picture film, and it typically allocates 10 bits of color depth for each RGB channel. The Cineon for-

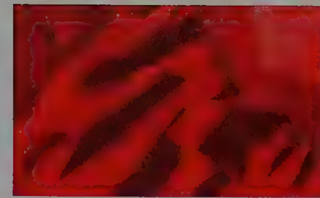
mat also offers a unique color space that excels at reproducing the subtle changes in density that exist on the original negative film which is the preferred scanning source. The **DPX** file format is a generic version of Cineon with the additional ability to store metadata information within its file header.

The **JPEG** file format, from **Joint Photographic Experts Group**, is one of the most popular 8-bit per color channel file formats that offers image compression. This is useful when large amounts of data have to be archived or transmitted over computer networks. JPEG works by removing on a frame by frame basis data that is redundant or data whose removal is almost imperceptible to the human eye. One of the main strengths of the JPEG format lies in the fact that it provides great compression and decompression speed as well as huge savings in file size (Fig. 15.3.3). The JPEG file format uses a lossy compression technique because it discards image information as it compresses the file. For this reason, the settings of the JPEG compression should be chosen with care, keeping in mind that increasing the compression decreases the image quality and vice versa. A copy of the original uncompressed file should always be archived in case the compression settings applied to the file were too extreme and the image becomes illegible. **JPEG 2000** is a newer version of this standard that uses **wavelet compression** techniques instead of discrete cosine transforms (**DCT**). A few of the improvements offered by JPEG 2000 include the lack of 8×8 pixel block artifacts, and a motion version.

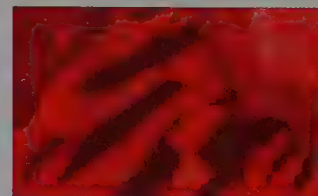
The **GIF** file format, from **Graphics Interchange Format**, is popular for compressing and storing images that are distributed on the Web—in fact, GIF was originally developed by the Compuserve online service. This file format is also compact enough to facilitate the uploading and downloading of still images over e-mail, Internet-based bulletin boards, and online services. The GIF format is based on an indexed color standard that ranges from 1-bit to 8-bit color depth for the three combined RGB channels and allows for a total maximum of 256 colors. These colors may be chosen from a customized color look-up table, but in some cases the colors of the system's color palette are automatically applied to GIF files. GIF also employs a lossless compression technique that requires a decompression utility before the file can be displayed on the receiving end.

The **PICT** file format, from the word picture, is a versatile format used by drawing and photo-retouching programs. PICT offers fair image quality with a relatively small file size. The PICT file format is used because of its compact size and compatibility across computer platforms. The **BMP** file format is another generic 8-bit per color channel bitmap format that is used in the Windows environment. The **PICS** file format derives its name from the word pictures, and is convenient when a series of images is required for animation purposes. A PICS file consists of several PICT files stored together.

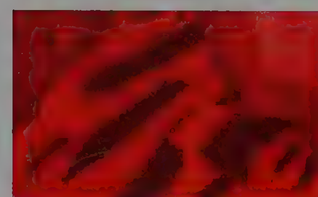
The **TGA** file format is popular with video-oriented software because it saves the files in an 8-bit per color channel format that is convenient for transferring the digital data into the video environ-



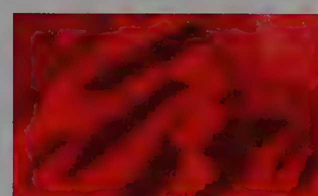
NO JPEG COMPRESSION, 556 KB



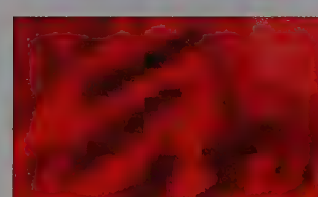
MAXIMUM QUALITY, 149 KB



HIGH QUALITY, 105 KB



MEDIUM QUALITY, 61 KB



LOW QUALITY, 39 KB

15.3.3 The JPEG compression options yield very different file sizes. Low compression results in higher image quality. The image degradation and blocking that results from maximum compression can be clearly appreciated in the enlarged detail at the bottom.

15.3.4 Computer animations saved in the QuickTime format have a playback controller at the bottom of the window.

ment. TGA is short for **TARGA**, the name of the family of graphics boards products developed in the early 1980s that pioneered video input and output with micro-computers.

The **OpenEXR** file format was initially developed as a proprietary format at Industrial Light & Magic but released as an open source format in 2003. OpenEXR was designed to better simulate the response of film negative to changes in exposure, and provides a higher dynamic range and color precision than existing 8-bit and 10-bit file formats. OpenEXR is based on a 16-bit floating point log color space, and can be used uncompressed or with several lossless image compression options that can deliver a 2:1 lossless compression ratio on images with film grain.

The **HDR** file format stands for High Dynamic Range and is based on the **Radiance RGBE** floating-point image format. High Dynamic Range files record a wide range of tonal detail, typically by merging into a single file a series of bracketed exposures of the same scene. The single HDR file contains the tonal detail of the merged exposures. HDR files are 32-bit and must be converted to 8-bit or 16-bit to be viewed in most systems.

Sequences of Moving Images

Some of the most popular file formats for saving sequences of two-dimensional images in a **self-contained format** include: QuickTime, QuickTime VR, MPEG, AVI, Windows Media, OMF, and AAF. These file formats are commonly used when the animated sequence is played back directly from a fast peripheral storage of the computer—as is the case with many interactive projects—or through computer networks. But sequences of images that are not played directly on the computer monitor, especially those destined to be recorded on film or videotape, are commonly saved in some of the same file formats used to save still images—mostly TIFF and TGA. In these cases, each individual sequential frame is assigned a number that reflects its place in the sequence.

15.3.5 Aspect ratios (width and height) of common formats in video (above) and film (opposite page), represented here with a fixed height of 1 for comparison purposes. Only relative proportions, not dimensions, are represented here.

The **QuickTime** file format stores both visual data and audio in one or several tracks. In most cases, a QuickTime file is displayed within a window that has a **playback controller,** such as those available in videotape players (Fig. 15.3.4). QuickTime is a versatile format that facilitates saving computer animation files at different spatial resolutions, and at different window sizes—for example, ranging from a small 160 × 120 pixels to a full-screen 640 × 480 pixels. QuickTime also plays the files back at different temporal resolutions—for example, 10 frames per second (fps) or 30 fps.

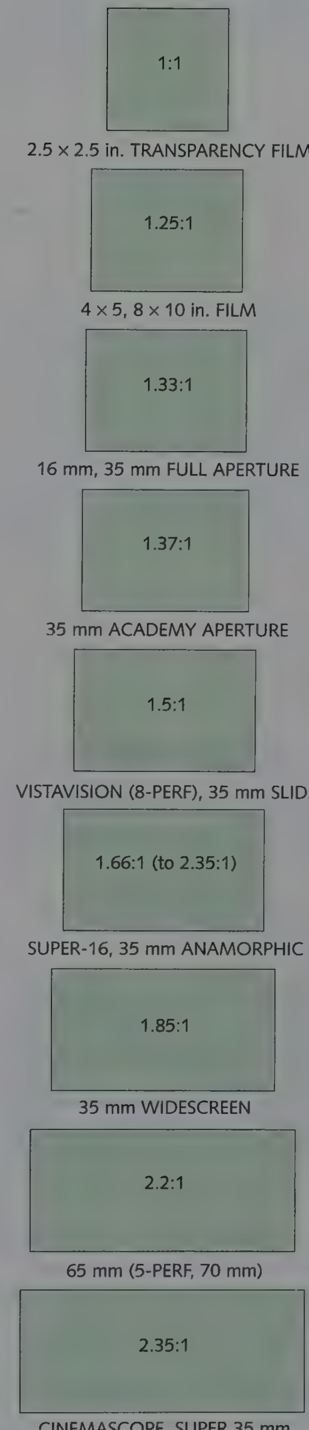
The QuickTime file format provides a variety of **compression options** for video or animated images. As with other compressed file formats, the quality and effectiveness of a QuickTime file is based on the relation between the compression ratio, the playback speed, and the image fidelity. The QuickTime file format provides several compression options. One of them is especially designed to compress

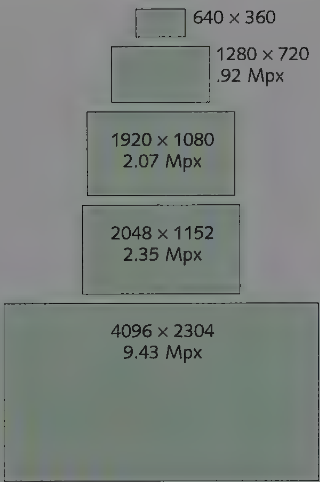
computer animation; another one is designed to compress photographic images and is based on the motion JPEG compression method. Some of the differences between each of these compression techniques are related to the time that it takes to compress and decompress an image. With some options the compression speed is high so that users do not have to wait long as the file is compressed but the compression ratio is low. With other options the compression ratio is good and that results in great space savings, but the compression time is slow. QuickTime is capable of online video streams and supports the **FireWire** (IEEE 1394) digital video serial interface, popular for video input/output from a computer system. QuickTime also provides sound sampling at different rates ranging from 8 to 48 kHz, and it is compatible with the popular MIDI format.

The **QuickTime VR** file format, also known as **QTVR**, creates panoramic views of real or virtual environments. This is done by assembling several still images that have been taken around the same point of rotation out toward the surrounding environment. The still images are stitched into a single panoramic image that represents a cylindrical view from the point of rotation. Once the QTVR file is assembled, the viewer can look in any direction and zoom in and out.

The **MPEG** file format is another popular format for compressed moving images, including video and animation. This file format was developed by the **Motion Pictures Experts Group**, which is affiliated with the International Standards Organization (**ISO**). The data compression in MPEG is based on the removal of data that is identical or similar not only within a frame but also between one frame to another. This lossy compression technique can realize impressive savings in file size while keeping a reasonable quality for most applications. MPEG files can only be displayed with utility programs called **MPEG viewers**, which usually provide a variety of dithering techniques for improving the final image quality. **MPEG-1** was initially developed to store moving images on Video-CD and for other types of low-bandwidth video compression. It provides a resolution of 352 × 240 pixels at 30 fps, which is roughly equivalent to the quality of VHS videotape. Incidentally, the **MP3** file format used to record music is an audio-only subset of MPEG-1; MP3 is the shortened name for MPEG-1 Layer III. **MPEG-2** is a higher-quality standard for high-bandwidth audio and video compression introduced in 1995. It offers a resolution of 704 × 480 at 30 fps, about twice the resolution of videotape. MPEG-2 requires dedicated hardware for playback, and it has become the compression standard for DVDs. **MPEG-4** is a newer standard issued in 2000 that offers added functionality. In addition to improved video and audio features, MPEG-4 has extensions for coding 3D polygonal meshes, and for defining and animating synthetic faces and bodies.

The **AVI** file format (**Audio Video Interleaved**) was introduced by Microsoft in 1992 as a generic digital format for moving images. Unlike the QuickTime format, AVI is not a cross-platform format, but AVI files can usually be converted and played by the QuickTime





15.3.6 The same aspect ratio, 16:9, showing relative size of different pixel resolutions ranging from Standard Definition video to a 4K digital file, the latter being 4096 x 2304 pixels or 9.43 Megapixels.

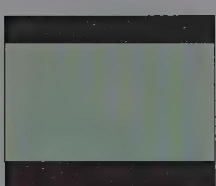
player. **Windows Media** is a more recent format developed by Microsoft that uses efficient compression for streaming material with high pixel resolutions. The **Flash**, or SWF, file format was developed by Macromedia and it handles graphics and animation. Flash files are usually more compact than GIF and JPEG because they can vector information instead of bitmaps. Flash also offers interactive capability and is popular with online applications such as the Web.

Two standards commonly used in postproduction for exchanging media between applications and platforms are OMF and AAF. The **OMF** file format, from Open Media Framework, offers 8-bit color per channel and different levels of lossy compression. It supports video, imbedded audio, effects, and edit decision lists (EDLs). OMF is used almost exclusively in nonlinear editing workflows based on Avid systems. The open source **Advanced Authoring Format**, or **AAF**, was designed to facilitate the authoring and postproduction processes. AAF is able to embed different types of content and authoring metadata information in the same *wrapper* while referencing back to the source files including video, animation, audio, and MIDI files among others. One of the strongest features of AAF is its ability to describe the process by which media in this format was created from the original source files. **MXF** is a related file format used to transport AAF content between systems.

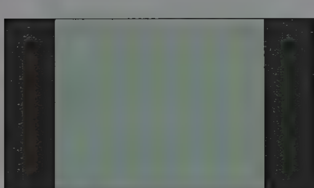
Aspect Ratios

Aspect ratios are not a component of file formats but both are part of the output process and, for that reason, they are closely related. **Aspect ratios** define the proportion between width and height in the active image area of a specific medium. Standard definition TV screens, for example, have a fixed aspect ration of 4:3 (4 wide by 3 high), while high definition video has a 16:9 aspect ratio. Film cameras offer a wide variety of recording aspect ratios to choose from, including 1.618:1, based on the classic golden section {1+sqrt(5)}/2, and 1.66:1, which is used to shoot the anamorphic version of 2.35:1. The numerical value of aspect ratios can be expressed in a variety of ways. TV screens, as mentioned before, have an aspect ratio of 4:3 that can also be expressed as 1.33:1. In the latter format the number 1 represents a fixed unit for the height while the width is a variable number; this format makes it easy to compare aspect ratios of different media. The 4:3 format is used because some people find it easy to remember due to its lack of decimal numbers. To make matters a bit more confusing, the 1.33:1 format is often expressed as 1:1.33, but in this context both arrangements usually mean exactly the same thing. Figure 15.3.5 shows the aspect ratios (with a fixed height of 1) for some of the most popular film and video output formats.

Image aspect ratios are not to be confused with **pixel aspect ratios**. Most pixels are round, or square, but some are not. **Nonsquare pixels**, tall ovals actually, are common in NTSC digital video and their ratio is 10:11. Most software programs automatically



16:9 LETTERBOXED INSIDE 4:3



4:3 PROTECTED INSIDE 16:9

15.3.7 The 16:9 aspect ratio is sometimes letterboxed into a 4:3 screen to broadcast feature movies on standard definition TV (top). A show might be recorded on 16:9 aspect ratio but with the 4:3 aspect ratio "protected" for a potential standard definition release (bottom).

compensate for nonsquare pixels, but if uncorrected this discrepancy may impact the image aspect ratio. An aspect ratio can be output in a variety of sizes, the same way a 35 mm photographic negative can be used to create different sizes of prints on photographic paper. When rendering an image for a specific output media we can choose a variety of pixel resolutions for the aspect ratio depending on a variety of issues like desired visual quality, planned distribution medium, deadline, and budget available. Rendering final images in the 16:9 aspect ratio (1.77:1), for example, could be done at 4096 × 2304 pixels, 2048 × 1152, 1920 × 1080, 1280 × 720, or 640 × 360 (Fig. 15.3.6). Keep in mind that there are always minor aspect ratio and pixel count differences between software programs, regions of the world, and input and output devices. The aspect ratio of the image captured by a standard definition NTSC video camera, for example, is not identical to the aspect ratio of the image displayed on the TV screen: a few pixels are not displayed. This fact accounts for the small difference in aspect ratios assigned to NTSC video: 1.333:1 and 1.327:1. This disparity between the captured and the displayed or projected aspect ratio is common in all output media, film included.

When a format conversion between media is necessary, usually the aspect ratio is impacted. When a production is recorded at 16:9, for example, there are usually two ways to show it in a 4:3 delivery medium such as a standard definition TV (16:9 is the native aspect ratio of HD cameras). The first method consists of cropping the extremes of the 16:9 image to accommodate the narrower aspect ratio of 4:3, assuming the recording was "protected" for a 4:3 release (most of the action happening in the center of the frame). The second method, called letterbox, consists of fitting the width of the 16:9 material to the width of the 4:3 delivery medium. The latter solution results in a smaller image and two empty black areas above and below the image area, but has the advantage of preserving the original composition and showing camera moves in their original form (Fig. 15.3.7).

15.4 Output on Paper

There is a great variety of printing technologies used for creating digital prints on paper. These include electrostatic, dye sublimation, ink jet, and pen plotters. Each of these printing technologies has strengths and weaknesses in the areas of resolution, paper size (Fig. 15.4.1), chromatic range, dye stability, and cost. The software required to control a specific type of printer is called a **printer driver**. The "preparation" for the printing of an image sent from a computer to a printer is often called **ripping**, from raster image processing. Printers often are as good (or as bad) as their RIP engines.

Electrostatic Printing

Electrostatic output technology is the most popular method for creating medium resolution black and white prints, but is also used to cre-

Metric ISO A Paper Sizes

Size	Millimeters	Inches
AO	841 × 1,189	(33.11 × 46.81)
A1	594 × 841	(23.39 × 33.11)
A2	420 × 594	(16.54 × 23.39)
A3	297 × 420	(11.69 × 16.54)
A4	210 × 297	(8.27 × 11.69)
A5	148 × 210	(5.83 × 8.27)
A6	105 × 148	(4.13 × 5.83)
A7	74 × 105	(2.91 × 4.13)
A8	52 × 74	(2.05 × 2.91)
A9	37 × 52	(1.46 × 2.05)
A10	26 × 37	(1.02 × 1.46)

American ANSI Paper Sizes

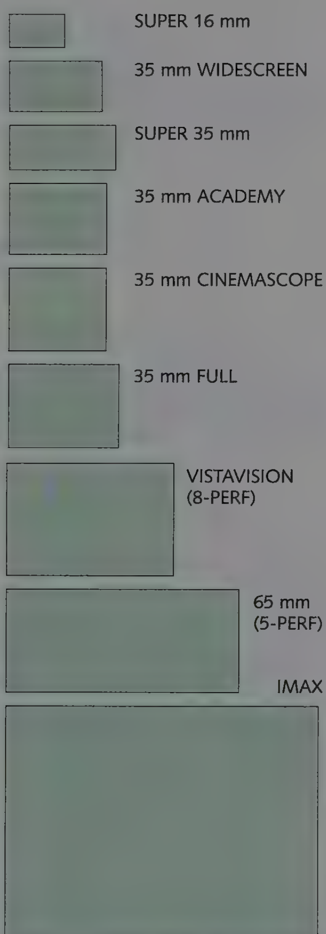
Size	Millimeters	Inches
A	8.5 × 11	(215.9 × 279.4)
B	11 × 17	(279.4 × 431.8)
C	17 × 22	(431.8 × 558.8)
D	24 × 36	(609.6 × 914.4)

Architectural Sizes

Size	Millimeters	Inches
A	9 × 12	(228.6 × 304.8)
B	12 × 18	(304.8 × 457.2)
C	18 × 24	(457.2 × 609.6)
D	24 × 36	(609.6 × 914.4)

15.4.1 International standard paper sizes used to specify the size of computer output on paper.

15.5.1 An ARRI laser film recorder. (© Arnold + Richter.)



15.5.2 Relative sizes of the major film aspect ratios (above and opposite page).

ate color prints. **Electrostatic** technology is commonly known as **laser printing**, and it is capable of resolutions that range from 300 to 1,000 dpi and higher. The output process with this technology typically consists of a laser beam that draws the image with electrical charges on a rotating metal drum, which in turn transfers charges onto the sheet of paper. Then the electrostatic energy on the paper attracts fine powder, or **toner**, to create an image directly on the surface of the paper. Finally, the toner is melted with heat on the paper. Many laser printers are able to print PostScript files and this further enhances the quality of their output, especially of line drawings. Electrostatic printing offers an affordable way of creating proofs on paper and heat-resistant acetate with sizes that range from letter size (8.5 × 11 in.) to tabloid size (11 × 17 in.).

Ink Jet Printing

Ink jet printers work like miniature airbrushes that spray microscopic drops of color ink on a sheet or a roll of paper. The nozzles through which the ink is sprayed are so thin that most of the ink dyes used in ink jet printers are based on vegetable dyes that have very small molecules. The small molecular size of the dyes allows the ink to pass through the narrow nozzles of the ink jet printers. However, another characteristic of vegetable dyes—as opposed to mineral dyes—is that the large majority are **fugitive dyes**, which means that they fade rapidly when exposed to the ultraviolet radiation of sunlight. This makes the ink jet printouts unstable unless they are coated with a transparent substance that acts as a filter of ultraviolet radiation—this is called **UV coating**. Using permanent dyes on acid-free paper with UV coating is the best way to create **archival-quality digital prints**. The strengths of ink jet printing include the excellent color and image fidelity, and the great availability of paper sizes and types. Ink jet printing also requires that RGB files be converted into the CMYK color format before they can be printed. In fact, most ink jet printers spray CMYK or CMY ink simultaneously on the paper, each color through a different nozzle. Of all the technologies for printing on paper that do not involve photographic processes, ink jet and dye sublimation are unique because of their good image and color fidelity.

Dye Sublimation Printing

Dye sublimation is a color printing technology that uses extreme heat to sublimate the dyes contained on a roll of acetate onto a sheet of paper. Sublimation happens to materials with such physical properties that when heated they are transformed from a solid state directly into a gaseous state, without passing through the liquid state. The sublimated dyes reach the paper in a gaseous state, and the pattern that they create on the paper is irregular but delicate, resembling the shape and pattern of the grains in photographic emulsions. For this reason, and also because of the pearl finish of the paper

used in dye sublimation, prints created with this technology resemble traditional photographic prints on paper. The irregular pattern created by the sublimated dyes that reach the paper in a gaseous state also softens the regularity of the grid of pins (dpi) that apply heat to form the image. This softening helps to increase the apparent resolution of dye sublimation printers, which is usually around 300 dpi but looks more detailed. Dye sublimation prints also offer great dye stability and excellent color range, but their cost is still higher than other techniques, and the maximum paper size rarely exceeds 11 × 14 in. This printing technology usually requires that RGB files be converted into the CMYK color format before they can be printed.

Digital Video Recording Formats	Color Sampling	Color Quantization	Compression Ratio	Recording Rates (bps)	Tape /Disc
High-Definition (± 1920 x 1080 pixels)					
D6	4:2:2	10/8-bit	Uncompressed	995 Mbps	19 mm
HDCAM-SR	4:2:2	10-bit	2.7:1, MPEG-4	440 Mbps	1/2 in.
HD-D5	4:2:2	10-bit	4:1, DCT	235 Mbps	1/2 in.
HDCAM	3:1:1	8-bit	4:1, DCT	142 Mbps	1/2 in.
DVCPRO 100/D-7	4:1:1~2:2	8-bit	6.7:1	100 Mbps	1/4 in.
Standard Definition, and Quasi-High-Definition					
D1	4:2:2	8-bit	Uncompressed	180 Mbps	19 mm
Digital Betacam	4:2:2	10-bit	2:1, DCT-based	90 Mbps	1/2 in.
IMX	4:2:2	8-bit	3.3:1, MPEG-1	50 Mbps	1/2 in.
D9	4:2:2	8-bit	3.3:1, DV	50 Mbps	1/2 in.
Blu-ray 1080p	4:2:0~4:4	8-bit	MPEG-4, VC-1	48 Mbps	50 GB
HDV (1440 x 1080i)	4:2:0	8-bit	MPEG-2	25 Mbps	1/4 in.
HDV (1280 x 720p)	4:2:0	8-bit	MPEG-2	19.7 Mbps	1/4 in.
DV (NTSC)/DVCAM	4:1:1	8-bit	5:1, DV	25 Mbps	1/4 in.
D-VHS	4:2:0	8-bit	MPEG-2	25 Mbps	1/2 in.
Betacam SX	4:2:2	8-bit	7:1, MPEG-2	18 Mbps	1/2 in.
DVD	4:2:0	8-bit	MPEG-2	3.5~9 Mbps	4.7 GB

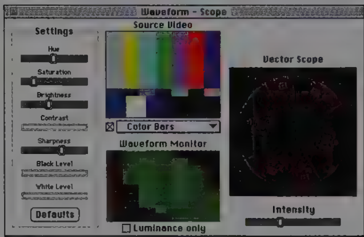
15.6.1 Some of the most popular component (not composite) video recording formats used in HD and standard definition production. A few notes: Recording rates are for video only, no audio. D6 records 1080i at 995 Mbps and 1080/24P at 796 Mbps. HDCAM prefilters the 1920 lines down to 1440 to minimize high-frequency noise; without prefiltering the compression ratio would be closer to 7:1. D9 was initially called Digital S. Blu-ray can be encoded at 4:2:0, 4:2:2, and 4:4:4, using MPEG-2, MPEG-4 AVC or VC-1. The color sampling rate for DV PAL is 4:2:0. The MPEG-2 exact compression ratio is a source of much discussion, it is estimated to be around 10:1.

Pen Plotters

Pen plotters have long been the preferred output technology for creating line drawings that do not require shading. Unlike other peripherals that also output onto paper, **pen plotters** do not create images with dots. Instead, they create drawings with continuous lines using one or several pens. As a result the image definition of line drawings created with pen plotters is excellent. The image definition of continuous tone images created with pen plotters, however, is quite limited because shading can only be simulated with cross-hatching patterns. Pen plotters are still quite popular in applications such as industrial design or architecture where drawings in very large formats, 36 × 48 in. for example, are necessary. High resolution electrostatic printers are an alternative to pen plotters when smaller size output is required.

15.5 Output on Photographic Media

In spite of the technological advances and innovations in the area of image output, photographic media are still the medium of choice when superb image quality and flexible size are required. The imaging devices that output on photographic media have a wide range of applications. Some are used to record stills on transparency film or photographic paper, or to record high-quality animation for projection in public theaters. Others are used to create and assemble images on high-contrast film for use in graphic arts mechanical reproduction.



15.6.2 A computer-simulated waveform monitor and vectorscope display with properly adjusted split field vertical color bars. (Adobe Premiere™ dialog box is reprinted with express permission by Adobe Systems Inc. Adobe and Adobe Premiere are trademarks of Adobe Systems Inc. or its subsidiaries and are registered in certain jurisdictions.)

Film Recorders

Film recorders are used to record computer images on both photographic film and paper. First-generation film recorders typically used a black and white monitor with filters for imaging on the film, but newer recorders use a laser beam to write directly on the film (Fig. 15.5.1). The high-end film recorders provide excellent resolution, chromatic range, and image permanence. Output on photographic film almost always requires that the images are sent to the recorder in the RGB color format. Film recorders for still images provide a convenient way to generate still images on transparency or negative film, and most accept the major film formats including 35 mm, 4 × 5 in., and 8 × 10 in. The film recorders used for motion picture film accept a variety of film formats, ranging from 16 mm to 35 mm.

Imagesetters

Imagesetters are high-resolution output peripherals that create black and white images. They are the preferred peripheral for outputting the films that contain the CMYK or spot color separations required for mechanical reproduction. (All the reproductions of three-dimensional imagery contained in this book, for example, were color separated, and output with digital imagesetters.) **Imagesetters** get their name from the photographic typesetters that were used to set type before digital technology transformed the typesetting industry. Imagesetters are capable of spatial resolutions in excess of 2,500 dpi, both for line art and for shaded images with halftone screens. Typically, a laser beam is used to draw the image directly on photosensitive film or paper, which is then developed and fixed with chemical solutions. The paper used in most imagesetters comes in rolls of different dimensions that accommodate the main paper sizes listed in Figure 15.4.1.

.aaf	Advanced Authoring Format
.avi	Audio Video Interleaved
.mpg	Motion Pictures Experts Group
.omf	Open Media Framework
.qt	QuickTime
.qtvr	QuickTime VR
.rm	Real Media
.swf	Flash

15.7.1 Some of the most popular file formats and their name extensions for moving images are listed here.

15.6 Output on Video

Videotape is the medium of choice for recording sequences of moving images intended for display on television sets, but video is sometimes also used to record still images. Output on video is commonly done on a wide variety of video formats, each one with its own peculiar advantages and limitations. This variety can sometimes complicate a task that in principle seems simple. A few of the video formats used are still analog, but the trend is toward digital video formats. Some of the former include Hi-8 mm; VHS, Betacam and D3 / in.; U-matic / in.; and D2 19 mm. The formats favored today for professional production are mostly digital component formats, and each one has a unique way of sampling, quantizing, and compressing the video information, as well as different recording rates (bits per second) and tape requirements. Figure 15.6.1 summarizes the main features of digital video recording formats.

Two of the most popular standards for encoding **Standard**

Definition (**SD**) video signals include NTSC and PAL. The NTSC signal is used in the United States, Japan, and most of Latin America. The PAL signal is used in most of the world, including most countries in Europe. The NTSC video standards were developed in the early 1950s, while PAL was developed almost 10 years later. NTSC has a resolution of 525 lines of information, but displays only 487 lines at the rate of 30 frames per second. The aspect ratio of NTSC is 4:3 (or 1.33:1). The PAL video standard displays 625 lines of information, but displays only 576 lines at the rate of 25 frames per second. PAL also has a unique automatic color correction system. Both NTSC and PAL use interlaced display of fields, but PAL does it at 50 Hz, while NTSC does it at almost 60 Hz.

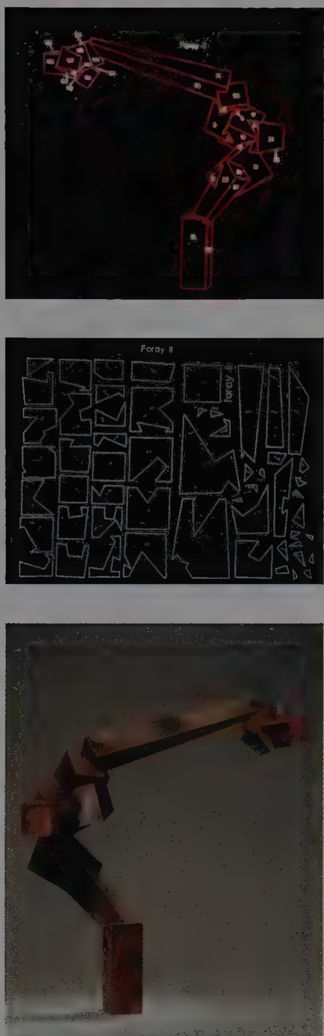
NTSC is a **composite video** signal that combines different types of information, including **luminance**, **chrominance** (color saturation and hue), and **timing sync information** (horizontal line synchronizing pulses, both horizontal and vertical blanking intervals, color reference burst, and reference black level). In **component video** the luminance and chrominance (or chroma) information remain as separate components usually referred to as Y, R-Y and B-Y; or Y (luminance), Pr and Pb (digitized color difference signal). An RGB video signal is also considered component video.

One of the **digital video** formats that are popular for independent productions or students' demo reels are **mini-DV** and HDV. The DV component format uses 1/4 in.-wide tape for recording standard definition video for both professional and consumer markets at a variety of rates including: 60i, 50i, 30p and 24p. DV is compressed to about 5:1, using a proprietary variant of DCT-based intrafield compression. The DV video recording rate is around 25 megabits per second. **HDV** is a format newer than DV, with a 4:2:0 color sampling rate and a 1440 × 1080i pixel resolution (or 1280 × 720 pixels in progressive mode), both higher than DV. The recording rate of HDV is 25 megabits per second for 1080i and 19.7 megabits per second at 720p.

The **High Definition** (**HD**) digital video format is larger and wider than standard definition NTSC or PAL. The native aspect ratio of HD is 16:9, and at 1920 × 1080 active pixels, the resolution is about twice that of standard television. A few additional rows of pixels are masked and inactive, and used to set the black level. There are a few major flavors of HD recording, each with different resolutions and frame rates: 1080/24p (which stands for 1080 active lines of progressive, non-interlaced, video at 24 fps, same rate as film); 1080/60i (1080 active lines of interlaced video at 60 fps); and 720p (720 lines of progressive video). Most 24p HD cameras capture color depth at 12-bit, output it at 10-bit, and sometimes record it at 8-bit (Fig. 15.6.1). To make things a bit more complicated, these HD formats used for recording images can be broadcast at slightly different resolutions depending on the country and TV network. MPEG-2 image compression is commonly used when the signal is transmitted to homes.

Theoretical Speeds [illegible]	
56K Modem	56 Kbps
ISDN (Integrated Services Digital Network)	128 Kbps
BRI (Basic Rate Interface)	384 Kbps
A-DSL (Asynchronous Digital Signal Level)	384 Kbps to 1.544 Mbps
T1	1.544 Mbps
10-Base Ethernet	10 Mbps
100-Base Ethernet	100 Mbps
ATM OC-3 (Asynchronous Transfer Mode)	155 Mbps
ATM OC-5	622 Mbps
HiPPI (High Performance Parallel Interface)	Up to 100 Mbps
FDDI (Fiber Distributed Data Interface)	100 Mbps
Gigabit Ethernet	1,000 Mbps (1 Gbps)
Fiber Channel	1 Gbps

15.7.2 Theoretical speeds in bits per second (bps) of different computer network standards, some still in the experimental stage. Often the overhead of managing the data brings the actual throughput 5 to 10 times less than the theoretical bit rate. BRI is popular for video conferencing. HiPPI and fiber channel are used for short distances between computers and storage devices. T1's signal is not distance-sensitive, while DSL suffers from data attenuation—its speed depends on the distance between the server and the computer. FDDI's popularity is fading.



15.8.1 *Foray* is a bronze sculpture created by Bruce Beasley, who uses computers to both compose and fabricate his sculptures. On top is the wireframe visualization, in the middle are the flat patterns to be cut in metal, and the finished piece is at the bottom. (Photograph by Lee Fatherree. Courtesy of Bruce Beasley.)

The quality of the final video output is largely determined by the video format, the quality of the videotape, and the video recording equipment. The quality of video output is also influenced by the proper balance and correction of RGB colors before they are recorded onto video, and by the calibration of the gamma factor. As mentioned earlier, the chromatic range of the RGB color model exceeds the range of video media, and often images in the RGB format contain colors that are outside the chromatic range of video and that create severe color distortions when displayed on video. For that reason, it is necessary to prepare images that were created in the RGB format before they are output to video. This is achieved through a **color correction** process that clips—or removes—the RGB colors that exceed the chromatic range of video, and replaces them with the closest equivalent color within the chromatic range of video (Fig. 14.2.5).

Another technical detail that has major consequences when outputting computer images onto video is the calibration of the **gamma factor**. The gamma factor—also known as gamma—makes the video image look as close as possible to the original RGB information by compensating for the loss of information between the voltages sent by the computer to the monitor, and the amount of light output by the monitor (Fig. 15.2.5). Other tools that are also useful in monitoring the quality of the video signal include the **vectorscope** and the **waveform monitor**. Both of these devices—whether real or simulated with software—display a graphical representation of the video signal. These graphs help to make sure, for example, that the colors fall within "legal" limits or that they are distributed evenly throughout the color spectrum, or that the transitions between color gradations are smooth and constant (Fig. 15.6.2).

Flickering is another common problem that occurs when computer-generated images are recorded onto NTSC video. **Flickering** happens for a variety of reasons but mostly due to the fact that each frame of NTSC video is displayed as two interlaced fields. One field contains all the even scan lines in the frame, and the other field contains the odd scan lines. RGB monitors are usually **noninterlaced**, so flickering occurs when a computer-generated image contains visual data such as horizontal lines or textures that are only one pixel high. This happens because that information appears only on one of the video fields and not on the other one (using antialiasing techniques can greatly reduce this problem). Flickering also happens when regular video cameras are used to record from an RGB monitor. This happens because the **refresh rate**, also called the scanning rate, of most NTSC video equipment is 60 Hz, while most RGB monitors usually have a faster rate. This type of flickering results in a diagonal line that rolls continuously from the top to the bottom of the video screen.

The aspect ratio of video has to be taken into consideration when recording computer-generated images onto videotape. Not only is the aspect ratio of video different from the aspect ratio of several RGB monitors, but also up to 20 percent of the image is cut off when computer animation is displayed on television video monitors.

(This has to do with the fact that the video signal uses some of the scan lines at the bottom of the screen to carry sync information.) An easy way to avoid having computer animations cut off is to preview them on a TV monitor and to use the field guides that specify the video **action safe areas** (Fig. 7.2.2).

15.7 Output on Digital Media

A considerable amount of three-dimensional computer renderings are delivered today in formats that can be readily used in digital media such as videogames or multimedia presentations. Professionals from a wide variety of visual disciplines are increasingly working with three-dimensional computer graphics in a **digital creative environment.** The traditional fields of graphic arts, broadcasting, and film each use three-dimensional creations in a specialized way, but they are all able to share their imagery in the form of **digital information**.

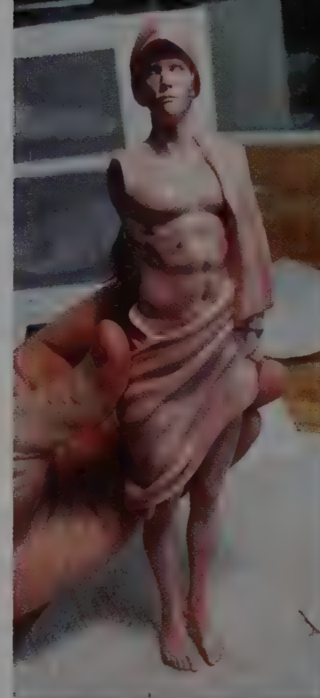
Three-dimensional images are output on digital media when they need to be played directly on the RGB monitor directly from the computer. This is the case, for example, when an animator wants to preview a motion test in the form of a sequence of low resolution files, or flipbook, stored in the hard disk. Other examples include the distribution of three-dimensional imagery and animation on optical media or through computer networks.

One of the main concerns with images delivered in digital media is that they are compact enough to load fast, but image detail is also important. To find the right balance between speed and detail it is necessary to perform tests that compare the amount of image detail against the loading speed and storage efficiency. Multiplayer online games, for example, cannot afford to use images with 16-bit color per RGB channel images, for this reason. The image quality would be superb, better than the standard 8-bit, but the loading speed would be too slow. The balance between speed and detail keeps changing as file formats become more sophisticated and graphics hardware more powerful and less expensive (Fig. 15.7.1).

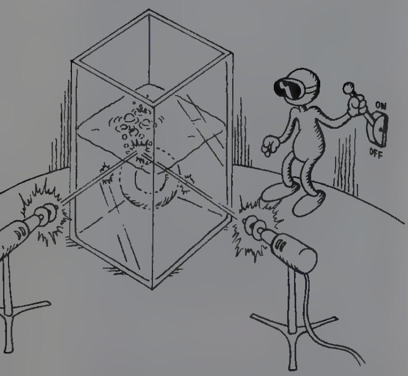
CD-ROM and DVD

CD-ROMs and DVDs are popular media for storing interactive content like games, educational materials, and feature films. Many of the file formats described earlier in the chapter are used in these media, with the preferred formats being the most compact ones.

Technically speaking, the **compact disk-read only memory**, commonly known as **CD-ROM** or **CD**, is a type of peripheral memory, but it can also be defined as output media because of its popularity for distributing visual projects. This medium is extremely convenient because it uses optical technology to store large amounts of information per disk, about 660 MB, in a format that is as permanent and stable as can be. Furthermore, the **read-time**—or time that it takes for the computer to find and read information from the CD-ROM—is



15.8.2 The finished model is cast in a fast-curing urethane tooling resin from the master pattern, detailed and painted. See Figure 15.1.1 for an earlier step in the fabrication process. (© 2002 Dan Platt, Solid Image Arts, LLC.)



15.8.3 The stereo lithography process hardens liquid polymer with two computer-controlled laser beams.

15.8.4 (Opposite page) The top image shows a finished bronze casting of a sculpture designed with NURBS curves. The middle image is a stereo lithographic model 4 in. in diameter fabricated with a white cornstarch-and-water material, which was then infiltrated with cyanoacrylate resin to add strength. The bottom image shows the bronze casting that was made directly from the stereo lithography part using a slightly modified version of the traditional lost wax process: the stereo lithography material and wax both break down at about the same temperature. The small stubs will become air vents, and the main gate through which the metal entered the casting can be seen in the lower left corner of the image. (© 1999 Bathsheba Grossman.)

minimal due to the laser beam technology used for this purpose.

CD-ROMs can contain all types of information. One CD-ROM, for example, may be filled up with over 60 minutes of very high-quality audio sampled at 44 KHz and in 16-bit stereo format, or with 450 images in 24-bit color RGB format at 640×480 pixel resolution. The same CD-ROM could also be filled up with over 20 hours of low resolution audio sampled at 11 KHz and in 8-bit stereo format, or with 26,500 monochromatic images at 512×342 pixel resolution. About 70 minutes of encoded MPEG-1 video fill up one CD-ROM.

The optical **DVD**, or **digital versatile disk**, is the same size as a CD-ROM but has a storage capacity larger than CD-ROM. There are a few versions of the popular **red laser DVD**. For example, a single-sided and single-layered DVD, called **DVD-5**, can hold up to 4.7 gigabytes of information. The single-sided and double-layered version, called **DVD-9**, can hold up to 8.5 GB of information. A **Video-DVD** is a variant of the DVD also popular for distributing high-quality video information like live action and animated feature films. DVDs use **variable bit rate** (VBR) MPEG-2 compression, multichannel audio, and subtitling capabilities. **Blu-ray disc** is a high definition DVD that can be encoded with MPEG-2, MPEG-4AVC (H.264), or SMPTE VC-1 codecs, and has a storage capacity of up to 50 GB on a dual-layer disc.

Network Downloading and Streaming

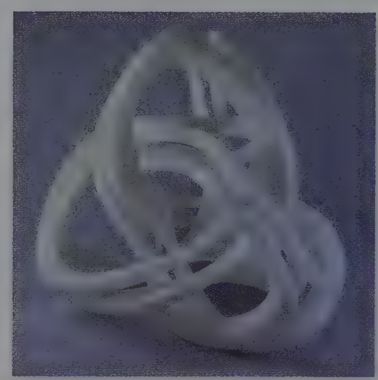
During the early days of the Internet it would have been adventurous trying to play computer animations in real time through a computer network. But advances in network technology today are contributing to make the distribution of computer-generated images and animations commonplace. The distribution of computer-generated images through a network serves different needs, and the system requirements in each case can have significant differences. A simple application of network distribution consists of playing back a sequence of animated images—sometimes called a **digital flipbook**—from one computer on the network directly on the monitor of another computer connected to the network and typically in the same room. There are two different strategies for sending animated sequences through a network: downloading and streaming.

Downloading involves copying a file through the network from the server onto another computer before it is played. The main advantage of downloading is superior playback quality because the file is being played locally. Another advantage is that the file can be saved for future use. However, downloading large files through narrow-bandwidth networks is time-consuming and often tedious. **Streaming** involves an almost instant playback of the file. Streaming files are played virtually as soon as they reach the local computer through the network. The great advantage of streaming is the ability to view files immediately, without having to wait until the entire file is downloaded. Disadvantages include the uneven playback quality

due to factors like network bandwidth and traffic, and the fact that streamed files cannot be saved in the local computer for future use. Standards for streaming three-dimensional data across networks are still evolving, and there are currently many competing standards.

In many computer animation production companies, digital files constantly travel through the **local area network**, or **LAN**. A computer that is dedicated to serving the requests of the network users is called a **server**, and many networks use multiple or **mirror servers** to increase efficiency. Animation files can be accessed by a **single user**, or simultaneously by **multiple users**. The latter would be the case if several individuals in different locations—a client in the meeting room and the production team in their studios, for example—needed to play back the same digital file as they spoke over the phone. A situation such as this one would benefit from a network with a wide bandwidth and/or the digital files having some form of data compression. There already are several networks in use that allow multiple simultaneous users to access a file with a minute loss in speed or resolution. Some of these solutions, however, require multiple servers and deep pockets.

The speed at which images are transmitted through networks of computers is an issue that influences the performance of playing back files remotely. The speeds at which files can travel through networks is related primarily to the bandwidth of the communication paths that files travel through between computers, as well as inside a single computer. These paths can be divided into three types: paths in a digital network, phone lines and modems, and paths inside the computer itself. The **bandwidth**, or transmission capacity, of these different media is measured with different units that express the amount of information that travels through the path per second. The bandwidth of digital networks is generally measured in **megabits per second** (Mbps). The bandwidth or speeds of modems operating on phone lines is measured in bits or **kilobits per second** (Kbps). The bandwidth of computer internal data channels is measured in **megabytes per second** (MBps). This seemingly arbitrary variety of bandwidth units is rooted in very different magnitudes of transmission capacity and in different traditions that date back to the early days of telephone transmission and computer engineering.



Planning the live playback of digital files through a network requires a wide variety of strategies depending on the purpose of the playback. Using internal networks, or **intranets**, can be different from using open-ended networks that go into the outside world. In the case of an intranet—a controlled network environment—it is possible to use specialized hardware for compression and decompression that increase the playback speed of an animation file through any network. In controlled network environments, each node can also be set up with specialized hardware and high bandwidth lines. In cases such as this, it makes sense to design the playback of the file by making extensive use of real-time file compression and decompression and high bandwidths. But in situations when the file is played in a wide variety of environments and throughout different bandwidths, it is

15.8.5 The finished *Medicus* physical sculpture (top). The figure was modeled as subdivision surfaces with LightWave software (below). (© 2002 Dan Platt, Solid Image Arts, LLC.)

usually wiser to choose the lowest common denominator so that the file can be played by the largest number of viewers. In any case, the number of products for software-based compression and decompression is increasing, as well as the resulting quality and performance.

The speed of digital networks is based on many factors, including the type of material that the network lines are made of (optical fiber or copper, for example), and the type of communications standards and protocol. As illustrated in Figure 15.7.2, the speed of modems and computer networks spreads a broad range. It is difficult to say that there is a standard network communications speed or bandwidth because this type of technology is still changing dramatically every couple of years. In the meantime, there are two key ideas to keep in mind. First, there is a significant difference in speed between the theoretical and the actual speeds of computer networks. This is due, in part, to the overhead of creating and managing the data packets that travel through the network; this can bring the actual throughput 5 to 10 times less than the theoretical bit rate! Second, data still travels inside the computer—and between the hard disks, RAM, and the computer—faster than it travels through computer networks.

The speed of internal data channels of specialized computers for visual creation is in the range of 5 to 80 megabytes per second. As a point of reference, a low rate of 10 megabytes per second traveling through the computer internal bus, for example, would be equivalent to a high rate of 80 megabits per second ($10 \times 1{,}000{,}000 \times 8 = 80{,}000{,}000$) traveling through a computer network. Compare that to the average range of telephone-based modem bandwidth: between 28,000 bits per second (28 Kbps) and 56,000 bps.

The total bandwidth and performance of file transmission over networks takes into account a multitude of factors, and cannot be determined by just one of the three data paths. In many instances, the performance of network distribution and playback of digital files is fast inside the computer, slow as the files are sent through the modem, and then fast again or very fast as the files travel through a network. For example, a file would travel fast inside a computer with an internal bus bandwidth of 40 megabytes (320 megabits) per second; it would then slow down as it goes through a 14,400 bits per second modem, zip through a high-speed optical fiber network at 135 megabits per second, slow down again through an standard Ethernet connection at 10 megabits per second, and play back even slower at a very slow computer with a bus bandwidth of only 2 megabytes (16 megabits) per second.

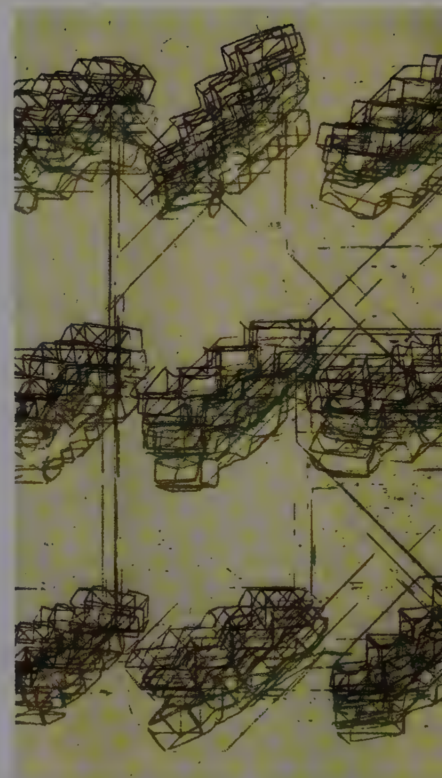
15.8 Output on Three-Dimensional Media

The data used to describe three-dimensional models in virtual environments can also be used in a variety of ways to actually build the object with real material. The details of the techniques used to translate digital three-dimensional data into real three-dimensional objects are beyond the scope of this book. But some of the basic techniques

are implemented with cutting machines, milling machines, and stereo lithography machines.

Computer-controlled **cutting machines** are able to cut two-dimensional shapes usually with a laser beam. These shapes are parts in structure that can be assembled by hand or under robot control. The shape of these parts is described as a series of XY coordinates or complex curves that are followed by the cutting tool. The work illustrated in Figure 15.8.1 was assembled from two-dimensional shapes created by unfolding a three-dimensional model with software designed for that purpose. This example was composed directly on the computer by previewing a large combination of shapes. All the planes in the sculpture were numbered and dimensioned on the computer screen. Once the model was fully resolved, the three-dimensional shapes were unfolded into two-dimensional patterns, plotted directly on foamcore, and a model was constructed. This model was sent to the foundry, and was burned out for a traditional lost wax casting. (If a sculpture is going to be fabricated rather than cast, the patterns are plotted on pattern paper for transfer to bronze plate.)

Milling machines, also known as subtractive rapid prototyping, are able to shape blocks of material such as plastic, wood, or stone by placing a rotating cutter on their surface, and moving across from top to bottom. The continuous motion of the cutting head along the three axes results in the modeling of a three-dimensional object. A wide variety of shapes can be created by using different cutting paths and by attaching different cutting tools to the milling machine (Fig. 15.8.6). The finished piece shown in Figure 15.8.5 was created by injecting hot toy wax formula into a flexible silicone mold previously created around parts machined on blue jewelers wax (Fig. 15.1.1). After cooling the wax reproduction is carefully removed and retooled by hand to remove imperfections and enhance delicate features. The cleaned-up pink figure becomes the master pattern that is molded again in silicone, recast in a fast-curing urethane tooling resin, finished, and painted (Fig. 15.8.2).



15.8.6 Detail of *Freedom and Imprisonment*, a traditional printmaking work partially created with computer-controlled engraving. The four-color vector paths of three-dimensional models were engraved with a needle directly on the copper plates used to make the print. (© Isaac Kerlow.)

Stereo Lithography

Stereo lithography is a process by which a liquid plastic or resin is shaped and solidified by two computer-controlled laser beams. The beams are perpendicular to each other and they move according to the XYZ positions on the surface of an object that has been modeled with software but is yet to be built in the physical world. One laser beam is focused at the transparent container from the front, and the other beam is focused from the side. The liquid plastic solidifies where the two laser beams intersect in the liquid inside the container (Fig. 15.8.3). With stereo lithography, shell surfaces or solid objects can be built by using the data contained in the digital model file to direct the motion of the two laser beams. Figure 15.8.4 shows three steps in the process of converting the model rendered in Figure 3.6.6 into a real bronze sculpture.

Chapter 15

Key Terms

2K files
3 to 2 (3:2) pull-down
4:4:4 color
4K files
Action safe areas
Advanced Authoring Format, AAF
Archival-quality digital prints
Aspect ratios
Audio Video Interleaved
AVI
Bandwidth
Binary system
Bitmap
Bitplane
Blu-ray disc
BMP
CD, CD-ROM
Chromatic range
Chrominance
Cineon
CMYK four-color separation
Codecs
Color bleeding
Color calibration
Color clipping
Color correction
Color depth
Color resolution
Color sampling rate
Compact disk-read only memory
Component video
Composite video
Compression options
Convert colors
Cutting machines
DCT
Device-independent file format
Digital creative environment
Digital flipbook
Digital information
Digital-to-analog
Digital versatile disk
Digital video, DV
Dot resolution
Dots per inch, dpi
Downloading
DPX
DTAs
DVCAM
DVD
DVD-5, DVD-9
Dye sublimation
Electrostatic
Encapsulated PostScript
EPS
Export filters
Export tools
File format conversion
File formats
Film recorders
FireWire
Flickering
Frames
Frames per second (fps)
Fugitive dyes
Gamma factor
GIF
Graphics Interchange Format
Graphics memory
Halftone output
Halftone screens
HDR
Header information
High Definition, HD
Image compression
Image definition
Image file
Image resolution
Imagesetters
Import filters
Import tools
Ink jet printers
Interlaced fields
Intranets
ISO
Joint Photographic Experts Group
JPEG, JPEG 2000
Kilobits per second
LAN
Laser printing
Lines per inch, lpi
Local area network
Lossless compression
Lossy compression
Luminance
LZW
Megabits per second
Megabytes per second
Milling machines
Mini-DV
Mirror servers
MPEG, MPEG-1, MPEG-2, MPEG-4
MPEG viewers
Motion Pictures Experts Group
Multiple users
MXF
Native file formats
Noninterlaced fields
Nonsquare pixels
NTSC
National Television Systems Committee
OpenEXR
Output peripherals
PAL, Phase Alternation by Line
Pen plotters
Peripheral device
PICT
Picture files
Pixel
Pixel aspect ratios
Pixel resolution
Pixels per inch
Playback controller
Portable image file formats
ppi
Printer driver
Quantization
QuickTime
QuickTime VR, QTVR
Radiance RGBE
Read-time
Red laser DVD
Refresh rate
Resample
Ripping
SECAM
Self-contained format
Series of still images
Servers
SGI
Single user
Spatial resolution
Standard Definition, SD
Stereo lithography
Streaming
Tagged Image File Format
TARGA
Temporal resolution
TGA
TIFF
Timing sync information
Toner
Translation of data
Universal image file formats
UV coating
Variable bit rate
VBR
Vectorscope
Video-DVD
Waveform monitor
Wavelet compression
Windows Media

Index

(Previous page: *Kung Fu Panda's* Shifu uses proven teaching techniques to make Po realize that he is a kung fu master. Notice the clear silhouettes, telling body language, and easy-to-read facial expressions.)

(Facing page: *Office Noise* characters. © 2008 The Animation Workshop, and Karsten Madsen, Mads Herman Johansen, Laerke Enemark, Torben S. Christensen.)

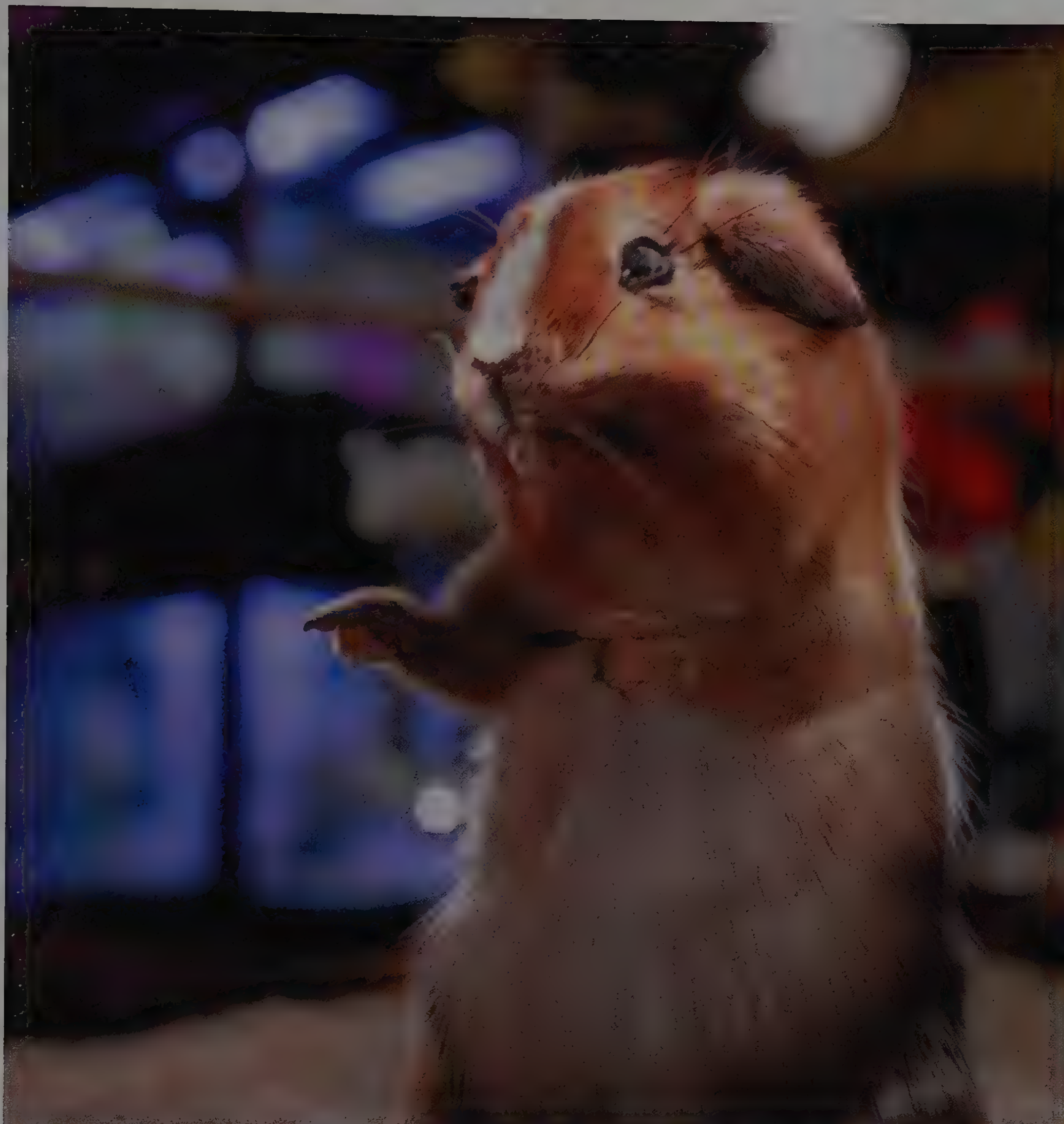
(*Madagascar: Escape 2 Africa*™ and © 2008 DreamWorks Animation LLC, used with permission. Previous page: *Kung Fu Panda*™ and © 2008 DreamWorks Animation LLC, used with permission.)

Index

• D •

E

F

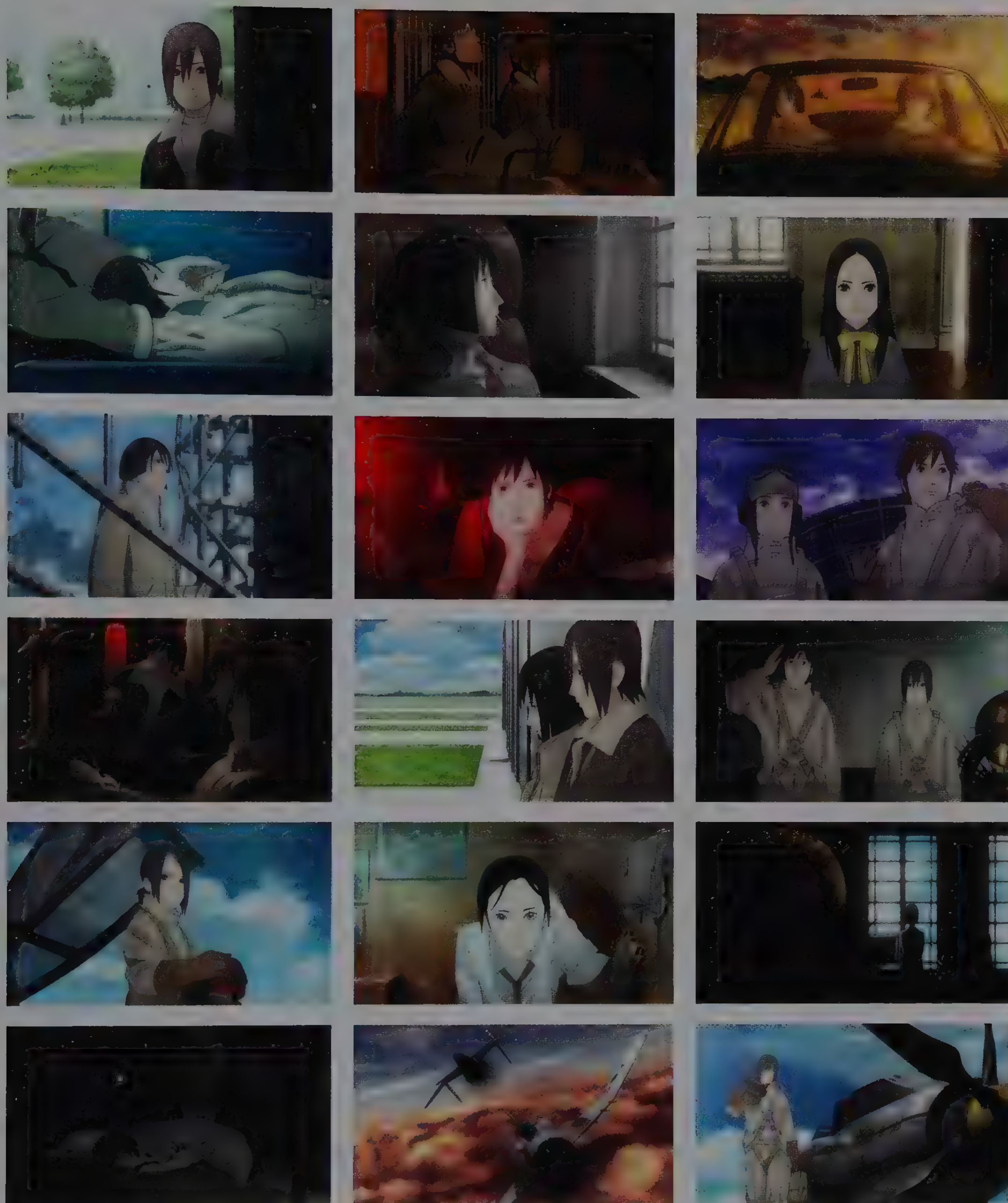


(Still frame from a commercial featuring *Ray*. © 2002 Blockbuster Entertainment/Tippett Studio.)

• M •

• Q •

• R •



(The two-dimensional characters in *Sky Crawlers* are composited over three-dimensional backgrounds and props. Related images in Figure 11.6.1. Images courtesy of Polygon Pictures. © MH/NI, BWDVYHDYCH.)

www.artof3d.com

(Top: *For the Birds*. © Disney Enterprises, Inc./Pixar Animation Studios.)

Visit this website to download selections from the *Timeline of Animation and Visual Effects* and the *Ten Career Tips for Digital Artists*.

Book is set in 10 on 11 Columbus Monotype, with headings and captions in Syntax.

Computer Animation/Visual Effects

Master the art of computer animation and visual effects production with the latest edition of this cutting-edge guide

This remarkable edition of *The Art of 3D Computer Animation and Effects* offers clear, step-by-step guidelines for the entire process of creating a fully rendered 3D computer animation. With up-to-date coverage of the latest computer animation styles and techniques, this versatile guide provides insightful information for creating animations and visual effects—from creative development and preproduction to finished animation.

Designed to work with any computer platform, this *Fourth Edition* cuts through technical jargon and presents numerous easy-to-understand instructive diagrams. Full-color examples are presented—including VFX and animated feature movies, games, and TV commercials—by such leading companies as Blue Sky, Blur, BUF, Disney, DreamWorks, Electronic Arts, Framestore, ILM, Imagi, Microsoft, Mac Guff, The Mill, Menfond, Pixar, Polygon, Rhythm & Hues, Sony Imageworks, Tippett, Ubisoft, and Weta, and many other studios and groundbreaking independent artists from around the world.

- Unique focus on creative development and production issues
- Non-platform specific, with multiple examples illustrated in a practical, step-by-step approach
- The newest computer animation techniques, including facial animation, image-based and non-photorealistic rendering, model rigging, real-time models, and 2D/3D integration
- Over 700 full-color images
- Encyclopedic timeline and production pipelines

This fully revised edition features new material on the latest visual effects techniques, a useful update of the traditional principles of animation, practical information on creative development, multiple production pipeline ideas for shorts and visual effects, plus updated information on current production trends and techniques in animation, rendering, modeling, rigging, and compositing.

Whether you are a student, an independent artist or creator, or a production company team member, *The Art of 3D Computer Animation and Effects, Fourth Edition* gives you a broad palette of tips and techniques for bringing your visions to life through 3D computer animation.

ISAAC KERLOW is a director and animation expert with over ten years of production and executive experience at The Walt Disney Company in Los Angeles, California. He is the Founding Dean of the first professional film and animation school in Singapore at the Nanyang Technological University, and a longtime active member of SIGGRAPH and the Visual Effects Society.

USA $65.00/CAN $78.00

Cover Illustration (top): A Gentlemen's Duel *Created by Blur Studio, Inc.*
Cover Illustration (bottom): Madagascar: Escape 2 Africa™ *and © 2008 DreamWorks Animation LLC, used with permission.*

Illustrations (back cover, from top to bottom): Même les pigeons vont au paradis. © *BUF, Director: Samuel Torneux.* Assassin's Creed © *Ubisoft Entertainment. All rights reserved. Assassin's Creed, Ubisoft, Ubi.com and the Ubisoft logo are trademarks of Ubisoft Entertainment in the U.S. and/or other countries.* Donkey Xote © *Donkey Xote S.A.—LUMIQ SPA—Castelao Productions S.A.—Bren Entertainment S.A.—Don Quijote de La Mancha 2005 S.A.* Coronado *image courtesy of Uncharted Territory, LLC.*

ISBN 978-0-470-08490-8
56500
9 780470 084908

WILEY
wiley.com